



TESIS TE142599

**ROBOT LENGAN PENGAMBIL BENDA UNTUK
MEMBANTU PASIEN DENGAN PERINTAH SUARA
MENGUNAKAN METODE MFCC DAN NEURAL
NETWORK**

Wahyu Muldayani
2213204011

DOSEN PEMBIMBING
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Tri Arief Sardjono, ST., MT

PROGRAM MAGISTER
BIDANG KEAHLIAN ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016



THESIS TE092099

OBJECT PICKER ARM ROBOT FOR PATIENT ASSISTANCE WITH VOICE COMMAND USING MFCC AND NEURAL NETWORK METHOD

WAHYU MULDAYANI
NRP. 2213204011

SUPERVISOR
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Tri Arief Sardjono, ST., MT

MAGISTER PROGRAM
FIELD STUDY OF ELECTRONIC ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tesis saya dengan judul **“Robot Lengan Pengambil Benda Untuk Membantu Pasien Dengan Perintah Suara Menggunakan Metode MFCC dan Neural Network”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2016

Wahyu Muldayani
NRP. 2213204011

**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)**
di

Institut Teknologi Sepuluh Nopember

Oleh:
Wahyu Muldayani
Nrp. 2213204011

Tanggal Ujian : 7 Januari 2016
Periode Wisuda : Maret 2016

Disetujui oleh:



1. Ir. Djoko Purwanto, M.Eng., Ph.D.
NIP: 19651211 199002 1 002

(Pembimbing 1)



2. Dr. Tri Arief Sardjono, ST., MT.
NIP: 19700212 199512 1 001

(Pembimbing 2)



3. Dr. Muhammad Rivai, ST., MT.
NIP: 19690426 199403 1 003

(Penguji)



4. Achmad Arifin, ST., M.Eng., Ph.D.
NIP: 19710314 199702 1 001

(Penguji)



5. Ronny Mardiyanto, ST., MT., Ph.D.
NIP: 19810118 200312 1 003

(Penguji)



Direktur Program Pascasarjana,

Prof. Ir. Djauhar Manfaat, M.Sc, Ph.D
NIP: 19601202 198701 1 001

ROBOT LENGAN PENGAMBIL BENDA UNTUK MEMBANTU PASIEN DENGAN PERINTAH SUARA MENGGUNAKAN METODE MFCC DAN NEURAL NETWORK

Nama Mahasiswa : Wahyu Muldayani
NRP : 2213204011
Pembimbing : 1. Ir. DjokoPurwanto, M.Eng., Ph.D.
2. Dr. Tri Arief Sardjono, ST., MT

ABSTRAK

Robot lengan pengambil benda dengan perintah suara adalah sebuah robot yang dapat digunakan untuk membantu manusia mengambil benda yang diinginkan dengan menggunakan perintah suara. Robot lengan ini diterapkan untuk membantu pasien yang memiliki keterbatasan gerak dalam mengambil benda yang diinginkan. Pengenalan perintah suara diproses menggunakan metode *MFCC* (*Mel-Frequency Cepstrum Coefficients*) dan ANN (*Artificial Neural Network*). Robot lengan juga dilengkapi kamera untuk mendeteksi benda yang akan diambil. Sensor ultrasonik diletakan pada ujung lengan robot untuk mengetahui jarak lengan terhadap target yang akan diambil. Pengenalan benda diproses dengan menggunakan metode *image-processing* berdasarkan warna, lebar dan tinggi pada benda. Limit switch diletakan pada salah satu lengan *gripper* robot digunakan sebagai tanda bahwa benda telah digenggam. Pada penelitian ini, robot lengan mampu mengambil benda yang diperintahkan menggunakan perintah suara dengan tingkat keberhasilan sebesar 78%.

Kata Kunci : MFCC, Robot Lengan, Suara.

OBJECT PICKER ARM ROBOT FOR PATIENT ASSISTANCE WITH VOICE COMMAND USING MFCC AND NEURAL NETWORK METHOD

Name : Wahyu Muldayani
NRP : 2213204011
Advisors : 1. Ir. DjokoPurwanto, M.Eng., Ph.D.
2. Dr. Tri Arief Sardjono, ST., MT

ABSTRACT

Object picker arm robot with voice command is a robot that can be used to help human to pick the object wanted using voice command. This arm robot was applied to help patient with movement disability to pick the object wanted. Voice command recognition was processed using MFCC (Mel-Frequency Cepstrum Coefficient) and ANN (Artificial Neural Network) method. The arm robot was also equipped with camera to detect the object. Ultrasonic sensor was placed at the end of the arm robot to measure the distance between the arm and the target. Object recognition was processed using image-processing method based on color, width, and height of the object. Limit switch was placed in one of the gripper arm of the robot and used as an indicator when the object was held. In this research, the arm robot was able to pick the object commanded using voice command with success rate of 78%.

Keywords : MFCC, Arm Robot, Voice

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH Subhanahu Wa Ta'ala atas limpahan nikmat Islam, Iman, ilmu dan kesehatan kepada penulis sehingga penulis dapat menyelesaikan tesis yang berjudul “Robot Lengan Pengambil Benda Untuk Membantu Pasien Dengan Perintah Suara Menggunakan Metode MFCC dan Neural Network”. Tesis ini disusun untuk memenuhi persyaratan dalam menyelesaikan Program Pascasarjana Teknik Elektronika di Institut Teknologi Sepuluh Nopember Surabaya.

Tesis ini tidak dapat tersusun dengan baik tanpa bimbingan, bantuan dan dukungan dari banyak pihak yang diberikan kepada penulis. Oleh karena itu penulis memberikan ucapan terimakasih dan penghargaan yang sebesar-besarnya kepada :

1. Ibunda dan Ayahanda yang selalu mendukung dan mendoakan hingga sampai terselesaikannya tesis ini.
2. Bapak Ir. Djoko Purwanto, M.Eng., Ph.D selaku dosen pembimbing yang telah banyak memberikan saran dan bantuan.
3. Bapak Dr. Tri Arief Sardjono, ST., MT selaku dosen pembimbing yang telah banyak memberikan saran dan bantuan.
4. Bapak Dr. Muhammad Rivai, ST., MT selaku dosen penguji dalam ujian tesis yang telah memberikan saran-saran yang sangat bermanfaat bagi penulis.
5. Bapak Achmad Arifin, ST., M.Eng., Ph.D selaku koordinator Bidang Studi Elektronika dan dosen penguji yang telah memberikan saran-saran yang sangat bermanfaat bagi penulis.
6. Bapak Ronny Mardiyanto, ST., MT., Ph.D selaku dosen penguji dalam ujian tesis yang telah memberikan saran-saran yang sangat bermanfaat bagi penulis.
7. Kepada Koordinator Pascasarjana Teknik Elektronika, Koordinator Pascasarjana Jurusan Teknik Elektro dan Ketua Jurusan Teknik Elektro serta karyawan Pascasarjana Teknik Elektro yang telah membantu penulis dalam segala urusan administrasi selama menempuh kuliah di ITS.

8. Rekan-rekan S1 dan S2 angkatan 2012, 2013, 2014, juga rekan-rekan Lab.B205, LabB202, dan LabB402 yang telah banyak membantu dalam menyelesaikan tesis ini.
9. Rekan-rekan pemain DOTA 2, Apis Cerana, Julismana, Kurome, Genius, Dapras, L2704RR, Pay, Kingsman, Laner, Vitamin A, Vitamin B, Vitamin C, Vitamin D, Vitamin E, Shina, Pukpuk dan Yonasumi yang telah menghibur ketika mengalami kesulitan dalam menyelesaikan thesis ini.
10. Rekan-rekan kos klampis aji 2 no 29 yang telah banyak membantu dalam menyelesaikan tesis ini.
11. Seluruh pihak yang membantu penulis dalam menyelesaikan tesis ini dengan baik.

Menyadari bahwa tesis ini masih jauh dari kata sempurna, maka kritik dan saran sangat penulis harapkan untuk perbaikan dimasa datang. Penulis berharap agar tesis ini dapat bermanfaat.

Surabaya, Januari 2015

Wahyu Muldayani
NRP. 2213204011

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan dan Manfaat Penelitian	3
1.4 Batasan Masalah	4
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1 Interpretasi Perintah Suara Menggunakan Fuzzy untuk Robot	5
2.2 Pelatih-Pemain Sistem Robot dengan Perintah Suara	6
2.3 Identifikasi Perintah Suara <i>Mel Frequency Cepstral Coefficient</i>	7
2.4 Pengenalan Perintah Suara	9
2.4.1 <i>Voice Activity Detection (VAD)</i>	9
2.4.2 <i>Mel-Frequency Cepstrum Coefficients (MFCC)</i>	10
2.4.3 <i>Neural Network</i>	15
2.5 Pengolahan Citra	19
2.5.1 <i>Color Image</i> atau RGB(<i>Red, Green, Blue</i>)	20
2.5.2 <i>Black and White</i>	21
2.5.3 <i>Binary Image</i>	22
2.6 Robot Lengan	23
2.7 Ziegler Nichols Untuk Tuning PID	24
2.8 Motor Servo	26
BAB 3 METODE PENELITIAN	29
3.1 Spesifikasi Robot Lengan	31

3.2	Pembentukan Perintah Suara Menggunakan MFCC dan NN.....	33
3.3	Sistem Identifikasi Target.....	42
3.4	Sistem Kendali Robot Mencari Target	46
3.5	Sistem Kendali Robot Mengambil Target	48
3.6	Integrasi Perintah Suara dan Robot Lengan dalam Mengambil Target..	51
BAB 4 HASIL DAN PEMBAHASAN		53
4.1	Pengujian Pembentukan Pola Perintah Suara Hasil Dari MFCC.....	53
4.2	Pengujian Sistem Identifikasi Pola dengan Neural Network.....	55
4.3	Pengujian Sistem Identifikasi Target.....	57
4.3.1	Pengujian Identifikasi Target Berdasarkan Warna Target.....	57
4.3.2	Pengujian Identifikasi Target Berdasarkan Tinggi dan Lebar Target.....	60
4.4	Pengujian Motor Servo	62
4.5	Pengujian Kecepatan Motor DC	63
4.6	Pengujian Robot Mencari Target.....	65
4.7	Pengujian Robot Mengambil Target.....	66
4.8	Pengujian Robot Mengambil Target dengan Perintah Suara.....	68
BAB 5 KESIMPULAN DAN SARAN		71
5.1	Kesimpulan.....	71
5.2	Saran	72
DAFTAR PUSTAKA.....		73
LAMPIRAN		77

DAFTAR GAMBAR

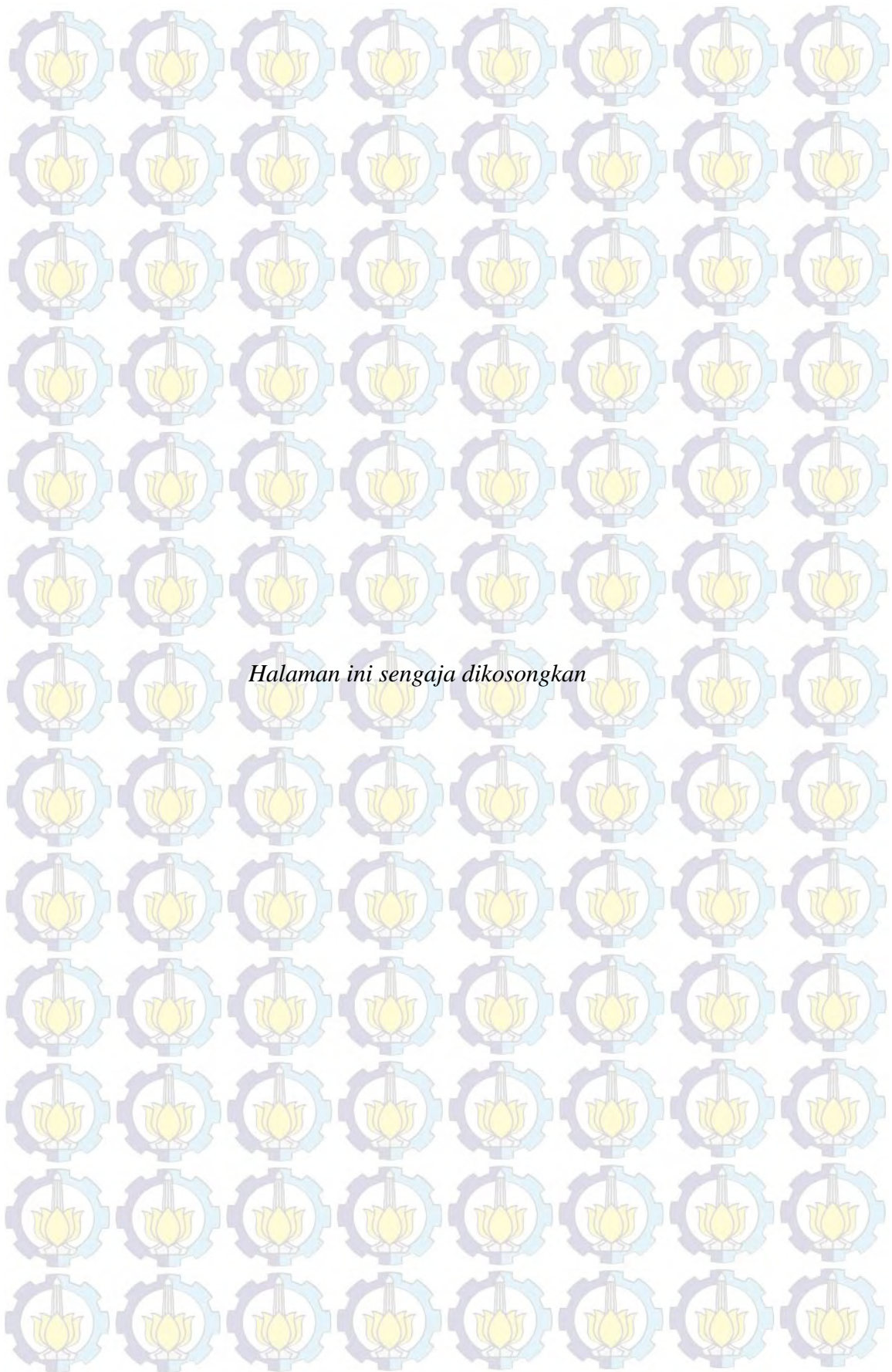
Gambar 2.1. Intrumentasi Peneliti jayasekara.	5
Gambar 2.2. Sistem Konfigurasi Keigo.	6
Gambar 2.3. Hasil Pemindahan Target Keigo.....	7
Gambar 2.4. Tahapan MFCC Dash	7
Gambar 2.5. Hasil Dari Pengenalan Suara Dash.	8
Gambar 2.6. Fishbone Diagram.....	8
Gambar 2.7. Block diagram of a VAD (Ramirez dkk, 2007).....	10
Gambar 2.8. Tahapan MFCC.	11
Gambar 2.9. Diagram <i>Butterfly</i> FFT.	13
Gambar 2.10. <i>Mel-Frequency Wrapping</i>	14
Gambar 2.11. Neural Network (Demuth dkk, 2009).....	16
Gambar 2.12. Struktur Dasar Pemodelan Neural Network.	16
Gambar 2.13. Color Image (Alasdair, 2004).	20
Gambar 2.14. Black and White (Grayscale) (Alasdair, 2004).	21
Gambar 2.15. Image Binary (Alasdair, 2004).	22
Gambar 2.16. Model Arm Robot 3 Dof Parekh.	24
Gambar 2.17. Kurva Reaksi.	25
Gambar 2.18. Reaksi sistem open loop ketika diberi input step.....	25
Gambar 2.19. Sistem Close Loop Motor Servo.	27
Gambar 2.20. Bentuk sinyal pengontrolan motor servo.....	27
Gambar 3.1. Ilustrasi Cara Kerja Robot Lengan Pengambil Target Mengguna-kan Perintah Suara	29
Gambar 3.2. Blok Diagram Proses Cara Kerja Robot Lengan Pengambil Target Mengguna-kan Perintah Suara	30
Gambar 3.3. Spesifikasi Robot Lengan.	32
Gambar 3.4. Blok Diagram Pembentukan Perintah Suara.	33
Gambar 3.5. VAD dari Kata "gelas merah".	35
Gambar 3.6. VAD Kata "gelas".....	35
Gambar 3.7. Pre-Emphasis Filter Kata "gelas".....	36

Gambar 3.8. FFT Kata "gelas".....	36
Gambar 3.9. Filter Bank dari Mel-Frequency Wrapping.....	40
Gambar 3.10. DCT Kata "gelas".....	41
Gambar 3.11. Topologi Neural Network.	41
Gambar 3.12. Blok Diagram Sistem Identifikasi Target.	43
Gambar 3.13. Target Warna Merah dengan Nilai R(150-256), B(0-80), dan G (0-80).	44
Gambar 3.14. Identifikasi Lebar dan Tinggi Gelas Merah.	44
Gambar 3.15. Posisi Set Point dan Titik Tengah dari Target yang Diambil Kamera.....	46
Gambar 3.16. Posisi Set Point dan Posisi Target Tampak Atas.	46
Gambar 3.17. Blok Sistem Kendali Robot Mencari Target.....	47
Gambar 3.18. Posisi Lengan Robot Terhadap Target yang Diinginkan 48	48
Gambar 3.19. Posisi Set Point Terhadap Titik Tengah Target Diambil Kamera 48	48
Gambar 3.20. Blok Sistem Kendali Robot Mengambil Target.....	49
Gambar 3.21. Posisi Ultrasonik Pada Lengan Robot Terhadap Jarak Target.....	50
Gambar 3.22. Posisi Ultrasonik Pada Lengan Robot Dengan Jarak Set Point Terhadap Target.....	50
Gambar 3.23. Flowchart Integrasi Perintah Suara dengan Robot Lengan.....	51
Gambar 4.1. Pola Keluaran MFCC dari Kata "gelas" (a) Pengucapan Pertama, (b) Pengucapan Kedua.	54
Gambar 4.2. Pola Keluaran MFCC (a) Kata "botol", (b) Kata "merah", (c) Kata "biru", dan (d) Kata "hijau"	54
Gambar 4.3. Pola Keluaran MFCC Kata "botol" dari Subjek yang Berbeda-beda.....	55
Gambar 4.4. Target Warna Merah dengan Nilai R(160-256), B(0-100), dan G(0-100).....	57
Gambar 4.5. Target Warna Merah dengan Nilai R(150-256), B(0-150), dan G(0-150).....	58
Gambar 4.6. Target Warna Merah dengan Nilai R(150-256), B(0-80), dan G(0-80).....	58

Gambar 4.7. Target Warna Hijau dengan Nilai R(0-100), B(0-100), dan G(100-256)	59
Gambar 4.8. Target Warna Biru dengan Nilai R(0-20), B(50-256), dan G(0-80)59	
Gambar 4.9. Identifikasi Lebar dan Tinggi Gelas Merah.....	60
Gambar 4.10. Identifikasi Lebar dan Tinggi Botol Merah.....	61
Gambar 4.11. Pengujian motor servo dengan lebar pulsa ON 1ms.....	62
Gambar 4.12. Pengujian motor servo dengan lebar pulsa ON 1.5ms.....	63
Gambar 4.13. Pengujian motor servo dengan lebar pulsa ON 2ms.....	63
Gambar 4.14. Pengujian Kecepatan Motor DC.....	64
Gambar 4.15. Grafik Respon Parameter Kontrol Robot Mencari Target.....	65
Gambar 4.16. Grafik Respon Parameter Kontrol Robot Mengambil Target dengan Jarak Target 15cm	67
Gambar 4.17. Grafik Respon Parameter Kontrol Robot Mengambil Target dengan Jarak Target 25cm	68
Gambar 4.18. Proses Pengujian Robot Mengambil Target dengan Perintah Suara.....	69

DAFTAR TABEL

Tabel 2.1. Hasil Dari Gerak Robot Terhadap Perintah Suara.	6
Tabel 2.2. Perbandingan Jumlah Perhitungan yang Dilakukan DFT dan FFT	13
Tabel 2.3. Tuning PID metode Ziegler-Nichols.....	26
Tabel 3.2. Input-Output Pembelajaran NN	42
Tabel 4.1. Hasil Pengujian Identifikasi Pola dengan Neural Network.....	56
Tabel 4.2. Hasil Nilai Min-Mak Identifikasi Target Berdasarkan Warna.....	59
Tabel 4.3. Hasil Identifikasi Target Berdasarkan Lebar dan Tinggi	61
Tabel 4.4. Hasil Pengujian Kecepatan Motor DC	64
Tabel 4.5. Hasil Pengujian Robot Mencari Target.....	66
Tabel 4.6. Hasil Pengujian Robot Mengambil Target.....	67
Tabel 4.7. Hasil Pengujian Robot Mengambil Target dengan Perintah Suara.....	70



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Robot bukanlah sesuatu yang hanya kita jumpai di dunia animasi atau film saja. Robot sudah menjadi kenyataan pada sekarang ini. Robot diciptakan manusia pada umumnya digunakan untuk mempermudah pekerjaan manusia. Hasil survei *United Nations* (UN) menghasilkan kelompok robotika dibagi menjadi tiga besar kategori yaitu: robotika industri, robotika layanan profesional dan robotika layanan pribadi (UN, 2002). Robot industri secara historis mewakili sebagian besar robot pembangunan dan banyak dikerahkan di otomotif industri. Professional robot layanan adalah robot yang bekerja otomatis tanpa ada perlakuan dari manusia seperti navigasi pada pertambangan dan membersihkan limbah nuklir. Robot layanan pribadi merupakan robot yang membantu orang secara interaksi langsung. Robot dapat berinteraksi dengan orang-orang yang tidak memiliki keahlian khusus atau pelatihan untuk mengoperasikan robot. Selama beberapa tahun terakhir, konsep robot sosial interaktif telah diperkenalkan dan menjadi salah satu topik terkini dalam penelitian di bidang interaksi manusia-robot (Bartneck dkk, 2004).

Interaksi manusia-robot memiliki cara yang berbeda-beda. Interaksi secara kontak langsung (Olabi dkk, 2010), manusia berinteraksi dengan robot dengan menekan tombol berupa keyboard atau joystick. Interaksi secara visual, robot berinteraksi dengan mengambil gambar atau keadaan sekitar. Seperti robot yang digerakan dengan mengambil gambar dari gerakan tangan manusia (Qian dkk, 2013). Interaksi manusia-robot dengan menggunakan suara, manusia dapat memerintahkan sebuah robot dengan menggunakan perintah suara (Ramdhani, 2009). Suara manusia sering digunakan sebagai alat komunikasi antar manusia. Manusia dapat berinteraksi memberikan perintah dengan menggunakan perintah suara. Jika suara ucapan digunakan untuk mengendalikan suatu robot, maka sistem yang dipakai harus berjalan secara realtime sehingga robot dapat dikendalikan secara interaktif (Ramdhani, 2009). Pada penelitian-penelitian

sebelumnya robot lengan digerakan menggunakan perintah suara hanya sebatas gerak naik, turun, kanan dan kiri (Jayasekara dkk, 2010).

Pada penelitian ini, robot lengan dibuat dengan kendali berupa perintah suara. Robot ini digunakan sebagai robot pelayan untuk membantu manusia dalam mengambil benda ketika mengalami keterbatasan gerak. Sebagai aplikasi robot perintah suara ini yaitu digunakan untuk membantu pasien dalam mengambil benda yang diinginkan. Robot ini akan membantu pasien yang mengalami keterbatasan dalam gerak, sehingga pasien dapat memperintahkannya dalam mengambil benda yang diinginkan dengan menggunakan perintah suara. Robot harus dapat membedakan suara sebagai perintah dan suara sebagai gangguan (noise). Pengenalan suara terdapat beberapa tahapan yang harus dilakukan, yaitu akuisisi data suara, ekstraksi ciri sinyal suara, dan pengenalan pola suara (Ramírez dkk, 2005). Proses akuisisi sinyal suara yang diperoleh oleh microphone akan didigitalisasi oleh PC. Proses ekstraksi ciri dan pengenalan pola suara terdiri dari tiga tahap, yaitu *Voice Activity Detection* (VAD), *Mel-Frequency Cepstrum Coefficients* (MFCC) dan *Neural Network*. Robot lengan juga dilengkapi camera untuk mendeteksi benda yang akan diambil sesuai dengan yang diperintahkan. Robot akan melihat warna, lebar dan tinggi benda untuk mengenali benda yang akan diambil. Dalam bidang robotik biasa disebut juga dengan vision robot. Vision robot dapat didefinisikan sebagai proses ekstraksi, karakteristik serta menafsirkan informasi dari dunia gambar tiga-dimensi. Proses ini disebut juga sebagai mesin atau vision komputer (*Computer vision*) dan dapat dibagi menjadi enam wilayah prinsip yang meliputi sensing, preproses, segmentasi, deskripsi, pengenalan dan interpretasi (Keigo dkk, 2006). Metode yang digunakan untuk membedakan warna, lebar dan tinggi benda yang akan diambil robot adalah metode pengolahan citra (*Image Processing*). Kamera pada robot akan mengambil gambar dari benda yang kemudian gambar tersebut diproses untuk menentukan warna, lebar dan tinggi benda. Setelah robot dapat mengetahui warna, lebar dan tinggi masing-masing benda, robot akan mengambil benda sesuai dengan yang telah diperintahkan. Pada penelitian ini, robot lengan mampu mengambil benda yang ditentukan dengan perintah suara secara real time guna mempermudah pasien.

1.2 Perumusan Masalah

Penelitian ini memiliki permasalahan yang harus diselesaikan, diantaranya :

1. Bagaimana robot lengan pengambil benda dapat membantu pasien yang memiliki keterbatasan gerak.
2. Bagaimana merubah suara menjadi data masukan sebagai perintah suara untuk robot lengan.
3. Bagaimana robot lengan dapat mengenali target yang diperintahkan.
4. Bagaimana robot lengan bergerak mencari target sesuai dengan yang diperintahkan.
5. Bagaimana robot lengan bergerak mengambil target sesuai dengan yang diperintahkan.
6. Bagaimana mengkombinasikan antara perintah suara dengan gerak robot lengan dalam mengambil benda untuk membantu pasien.

1.3 Tujuan dan Manfaat Penelitian

Penelitian ini memiliki tujuan, diantaranya :

1. Menghasilkan robot lengan pengambil benda untuk membantu pasien yang memiliki keterbatasan gerak.
2. Merubah suara menjadi perintah suara untuk menggerakan robot lengan.
3. Memandu robot lengan untuk mengenali target sesuai dengan yang diperintahkan.
4. Mengatur gerak robot lengan mencari target sesuai dengan yang diperintahkan.
5. Mengatur gerak robot lengan mengambil target sesuai dengan yang diperintahkan.
6. Mengkombinasikan suara dengan gerakan robot lengan dalam menentukan benda untuk membantu pasien.

1.4 Batasan Masalah

Penelitian ini memiliki batasan masalah, diantaranya :

1. Benda yang digunakan berupa gelas dan botol yang berbeda warna.
2. Benda yang digunakan berjumlah 5 benda.
3. Pengaturan pencahayaan dan warna sekitar.
4. Benda diletakan secara acak dengan letak yang ditentukan.
5. Suara lingkungan tidak terlalu bising.

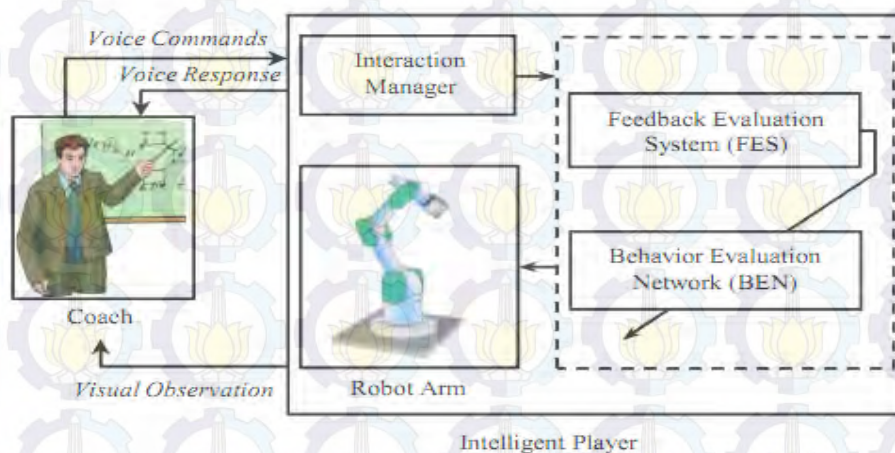
BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Perancangan Robot lengan pengambil dan peletakan benda dengan perintah suara untuk membantu pasien tidak lepas dari kajian pustaka dan teori dasar. Berikut adalah beberapa kajian pustakan yang berkaitan dengan rancangan penelitian ini.

2.1 Interpretasi Perintah Suara Menggunakan Fuzzy untuk Robot

Jayasekara dkk, melakukan penelitian perintah suara terhadap robot lengan dengan menggunakan metode fuzzy voice feedback. Tahapan penelitian yang dilakukan oleh jayasekara dapat dilihat pada Gambar 2.1. Sistem robot lengan berbasis suara yang dibuat menggunakan informasi linguistik fuzzy dalam perintah pengguna berdasarkan feedback suara fuzzy. Perilaku primitif dievaluasi oleh *behavior evaluation network* (BEN). Feedback evaluation system (FES) diperkenalkan untuk mengevaluasi umpan balik pengguna untuk memperbaiki persepsi robot dengan mengadaptasi BEN. Pada penelitian yang dilakukan ini suara hanya digunakan untuk menggerakkan robot bergerak keatas, kekiri, maju dan lainnya tanpa ada interaksi balik ke pengguna, seperti pada Tabel 2.1. Noise dalam suara juga tidak terlalu diperhitungkan dalam penelitian ini.



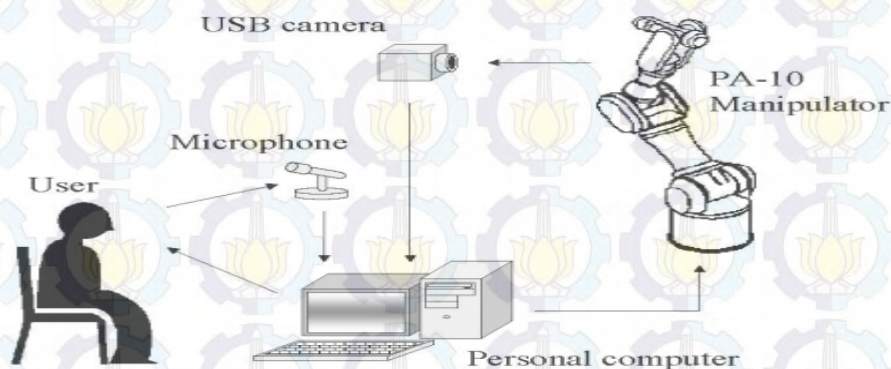
Gambar 2.1 Intrumentasi Penelitian jayasekara

Tabel 2.1 Hasil Dari Gerak Robot Terhadap Perintah Suara.

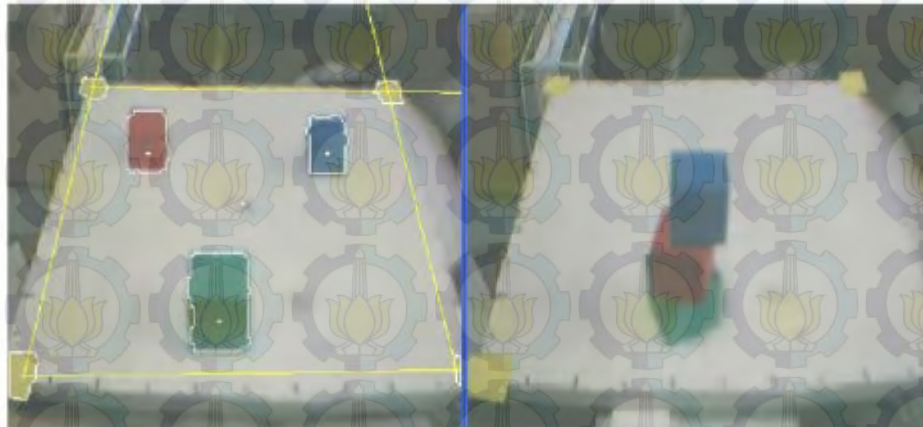
User command	Tip position [mm]			l/[mm]
	x	y	z	
(Point E)	463.44	-368.69	152.02	
Move up	463.37	-368.69	267.86	115.84
Move far left †	463.38	-202.81	267.82	165.88
Move far left	463.30	-13.99	267.69	188.82
Move far left	463.30	176.73	267.65	190.72
Move left	463.29	317.62	267.62	140.89
Move forward	572.08	317.76	267.73	108.79
Move forward	699.79	317.62	267.77	127.71
Move very little forward	735.99	317.64	267.79	36.20
Move little down	736.08	317.64	183.21	-84.58
Move very little down ‡	736.08	317.59	167.81	-15.40
Move very little up §	736.06	317.62	177.39	9.58

2.2 Pelatih-Pemain Sistem Robot dengan Perintah Suara Menggunakan Neural-Fuzzy

Watanabe Keigo dkk, melakukan penelitian perintah suara sebagai interaksi manusia robot. Metode yang diggunakan adalah sistem pelatih-pemain fuzzy, di mana manusia dianggap sebagai pelatih, robot dipandang sebagai pemain, dan manusia dengan suara dijadikan imputan implikasi fuzzy sebagai salah satu cara komunikasi. Sistem konfigurasi yang dilakukan keigo seperti pada Gmbar 2.2. Sebuah penalaran suara dideteksi menggunakan metode fuzzy neural network dalam menentukan input kontrol gerakan robot. Kamera diletakkan terpisah dengan robot. Kamera ini diggunakan untuk memantau target yang akan dipindahkan. Gambar 2.3 merupakan hasil dari pemindahan target yang dilakukan keigo. Namun dalam penelitian ini suara pengguna dalam keadaan normal tanpa adanya gangguan.



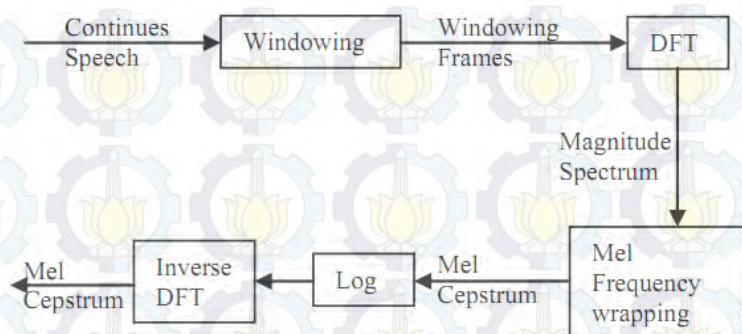
Gambar 2.2 Sistem Konfigurasi Keigo



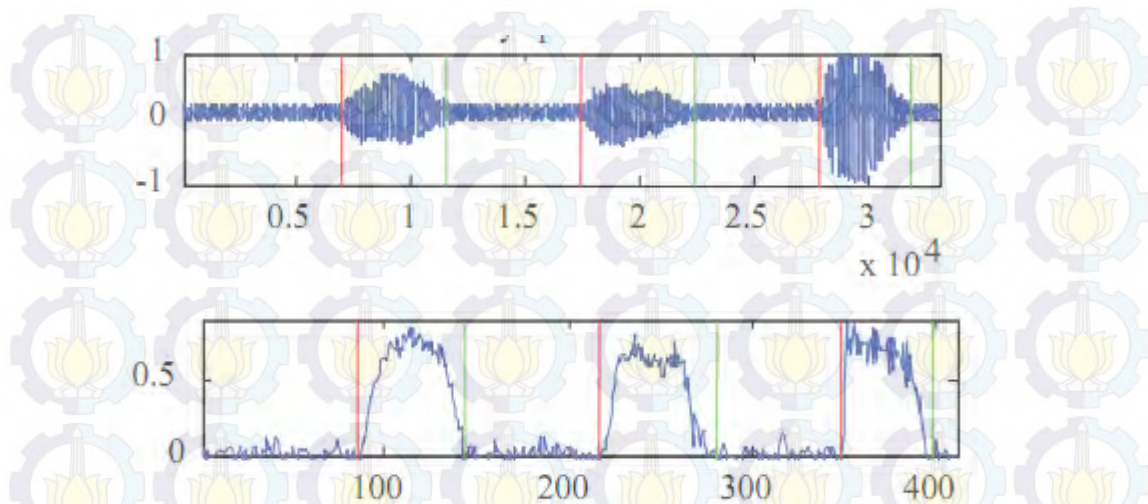
Gambar 2.3 Hasil Pemindahan Target Keigo

2.3 Identifikasi Perintah Suara Menggunakan Mel Frequency Cepstral Coefficient dan BPNN

Dash dkk, melakukan penelitian tentang identifikasi pengenalan suara. Metode yang digunakan adalah *Mel Frequency Cepstrum Coefficient* (MFCC) dan *Back Propagation Neural Network* (BPNN). MFCC digunakan sebagai fitur untuk merancang teks identifikasi suara yang bergantung pada sistem. BPNN digunakan untuk identifikasi suara setelah pelatihan set fitur dari MFCC. Beberapa modifikasi yang ada teknik MFCC untuk ekstraksi fitur juga disarankan untuk meningkatkan efisiensi pengenalan suara. Gambar 2.4 merupakan tahapan MFCC yang dilakukan oleh Dash. Gambar 2.5 merupakan hasil pengenalan suara yang dilakukan oleh Dash. Namun dalam penelitian yang dilakukan oleh Dash hanya terfokus pada pengenalan suara tidak terhadap kekebalan terhadap noise yang mungkin terjadi.

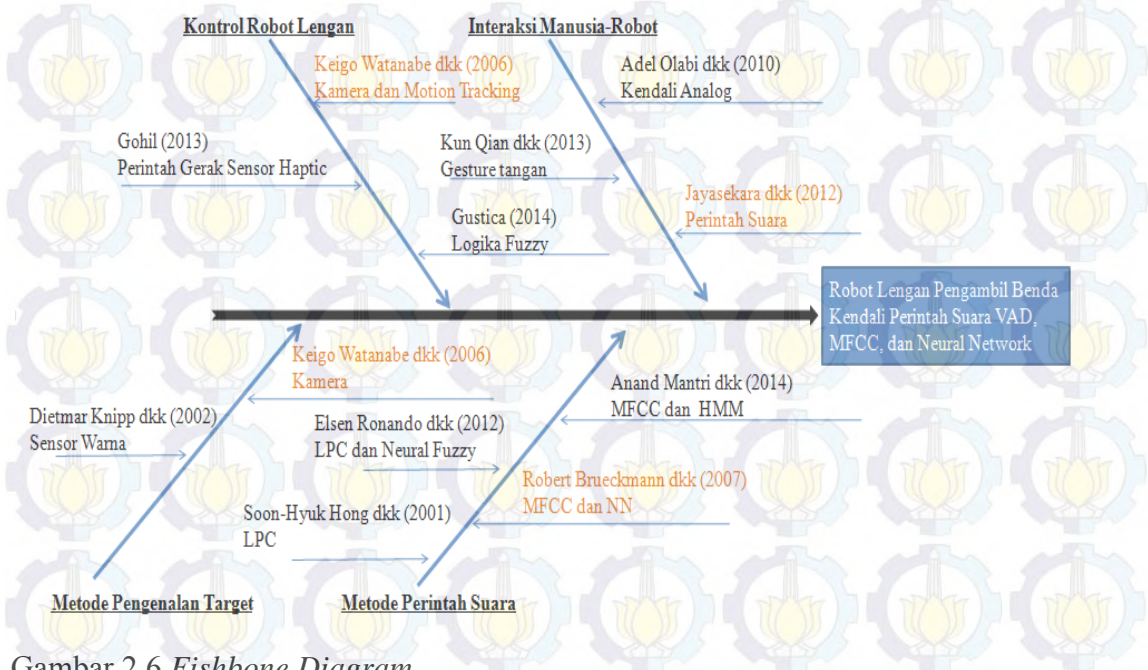


Gambar 2.4 Tahapan MFCC Dash



Gambar 2.5 Hasil Dari Pengenalan Suara Dash

Gambar 2.6 merupakan block fishbone diagram penelitian yang berkaitan dengan perancangan penelitian yang akan dibuat. Penelitian-penelitian terdahulu terbagi dalam empat golongan yaitu, kontrol robot lengan, interaksi manusia-robot, metode pengenalan target dan metode perintah suara. Kontrol robot lengan yang digunakan yaitu motion tracking menggunakan kamera. Interaksi manusia dengan robot menggunakan perintah suara. Metode pengenalan perintah suara menggunakan *Mel Frequency Cepstrum Coefficient* MFCC dan Neural Network. Pengenalan target dilakukan dengan menggunakan kamera.



Gambar 2.6 Fishbone Diagram

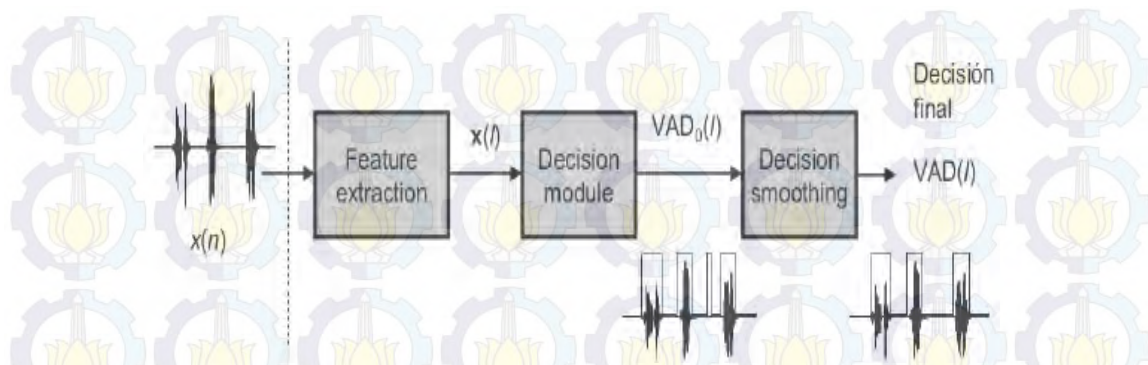
Dasar teori yang mendukung dalam perancangan penelitian ini adalah sebagai berikut:

2.4 Pengenalan Perintah Suara

Suara adalah sebuah sinyal yang rumit sebagai sebuah hasil dari beberapa transformasi yang terjadi pada beberapa level yang berbeda dari semantik, linguistik, artikulasi (pengucapan) dan akustik. Perbedaan dalam transformasi ini tampak sebagai perbedaan anatomic yang melekat dalam vocal dan kebiasaan pengucapan yang dipelajari dari individu yang berbeda (Rabiner dkk, 1993). Suara pada manusia digunakan untuk berkomunikasi atau berinteraksi antara manusia. Suara dapat digunakan sebagai data masukan sistem dengan menggunakan sensor suara. Sensor Suara adalah sensor yang memiliki cara kerja merubah besaran suara menjadi besaran listrik. Pada dasarnya prinsip kerja pada alat ini hampir mirip dengan cara kerja sensor sentuh pada perangkat seperti telepon genggam, laptop, dan notebook. Sensor ini bekerja berdasarkan besar kecilnya kekuatan gelombang suara yang mengenai membran sensor yang menyebabkan Bergeraknya membran sensor yang memiliki kumparan kecil dibalik membran tersebut naik dan turun. Kecepatan gerak kumparan tersebut menentukan kuat lemahnya gelombang listrik yang dihasilkannya. Salah satu komponen yang termasuk dalam sensor ini adalah Microphone atau Mic. Mic adalah komponen elektronika dimana cara kerjanya yaitu membran yang digetarkan oleh gelombang suara akan menghasilkan sinyal listrik. Dalam penerapan suara sebagai interaksi manusia terhadap robot lengan dalam rancangan penelitian ini terdiri dari tiga tahap, yaitu *voice activity detection* (VAD), *Mel-Frequency Cepstrum Coefficients* (MFCC) dan *Neural Network*.

2.4.1 Voice Activity Detection (VAD)

Suara terdiri dari suara sebagai informasi dan suara sebagai noise. Dalam mendeteksi suara informasi/suara noise merupakan masalah kompleks dalam proses pengenalan suara. VAD adalah sebuah metode untuk membedakan segmen suara informasi dan suara noise.



Gambar 2.7 Block diagram of a VAD (Ramirez, 2007)

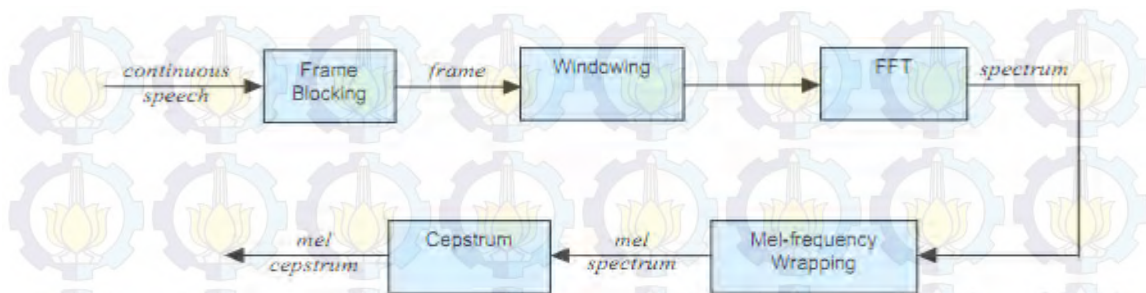
VAD sebagian banyak diaplikasi pada pemrosesan audio dan secara luas digunakan dalam bidang komunikasi suara untuk mencapai tinggi efisiensi dan bit rate rendah transmisi (Mousazadeh dkk, 2006). Contohnya termasuk pengurangan kebisingan untuk perangkat bantu dengar digital, layanan komunikasi bergerak, sistem pengenalan suara, kompresi, dan pidato coding. Voice Activity Detector adalah setiap kumpulan dari metode untuk segmen sinyal suara terdiri dari suara, tak bersuara dan noise. Secara umum, VAD memiliki tiga bagian komponen: *Feature Extractor*, *Decision Module*, dan *Decision Smoother* (Ramirez dkk, 2006a). Gambar 2.7 memberikan perancang peningkatan fleksibilitas dalam menciptakan solusi VAD. Metode yang digunakan adalah menghitung *power* sinyal dalam satu satuan waktu tertentu ditunjukkan pada Persamaan 2.1.

$$p = \frac{1}{n} \sum_{j=1}^N (x_j^2) \quad (2.1)$$

Dimana, p adalah *power* sinyal, x_j adalah sinyal wicara pada detik ke- j dan N adalah panjang moving average *filter*

2.4.2 Mel-Frequency Cepstrum Coefficients (MFCC)

Berbagai macam metode mewakili sinyal suara untuk pengenalan pola perintah suara, seperti *Linear Prediction Coding* (LPC), Mel-Frequency Cepstrum Coefficients (MFCC) dan lain-lain.



Gambar 2.8 Tahapan MFCC

MFCC merupakan metode yang paling dikenal dan paling populer. Suara direkam pada sampling rate diatas 10000 Hz. Frekuensi sampling ini dipilih untuk meminimalkan efek konversi di analog ke digital. Sinyal-sinyal sampel dapat menangkap semua frekuensi hingga 5 kHz, menutupi sebagian besar energi suara yang dihasilkan oleh manusia. Tujuan utama dari prosesor MFCC adalah untuk meniru perilaku telinga manusia. Gambar 2.8 menunjukkan blok diagram dari prosesor MFCC. Proses ini dibagi menjadi 5 bagian, yaitu *Frame Blocking*, *Windowing*, *FFT*, *Mel-Frekuensi Wrapping* dan *Cepstrum* (Leon, 2009).

a. *Frame Blocking*

Sinyal suara akan terus mengalami perubahan akibat adanya pergeseran artikulasi dari sumber suara, sehingga sinyal harus diproses dalam satuan waktu tertentu. Panjang *frame* yang digunakan adalah antara 20 - 30 milidetik. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Ukuran *frame* harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi di lain sisi, ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik. Proses *frame blocking* ini dilakukan terus sampai seluruh sinyal dapat diproses. Proses ini dilakukan secara *overlapping* untuk setiap *frame*-nya. Panjang daerah *overlap* yang digunakan adalah 25% hingga 50% dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*.

b. *Windowing*

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral atau *aliasing*. *Aliasing* adalah sinyal baru dimana memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek ini dapat terjadi karena rendahnya jumlah

sampling rate, ataupun karena proses *frameblocking* sehingga menyebabkan sinyal menjadi *discontinue*. Windowing merupakan suatu fungsi yang berfungsi untuk mengalikan sinyal terpotong yang *discontinue* dengan fungsi window agar menjadi sinyal *continue*. Windowing direpresentasikan seperti yang ditunjukkan pada Persamaan 2.2 dan 2.3.

$$Y[n] = X(n) * W(n) \quad 0 \leq n \leq N-1 \quad (2.2)$$

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (2.3)$$

Dimana $Y[n]$ adalah keluaran sinyal, $X(n)$ adalah masukan sinyal, $W(n)$ adalah Windowing dan N adalah jumlah sample tiap frame

c. *Fast-Fourier Transform* (FFT)

Suatu fungsi dengan periode tertentu dapat dinyatakan dalam deret fourier. Transformasi fourier digunakan untuk mengubah deret sinyal domain waktu berhingga menjadi domain frekuensi. Persamaan yang mendasari transformasi fourier adalah Persamaan 2.4 dan Persamaan 2.5 yang merupakan persamaan *discrete fourier transform* (DFT).

$$x[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad (2.4)$$

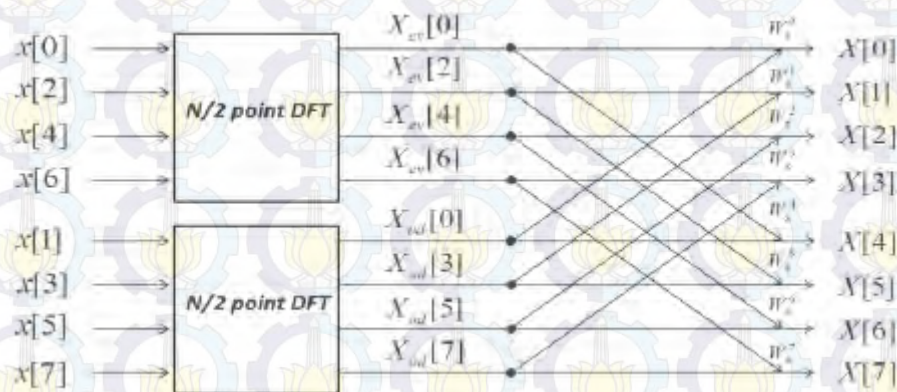
$$W_N^{kn} = e^{-j2\pi kn / N} \quad (2.5)$$

$x[k]$ merupakan pola frekuensi ke- k dan $x[n]$ merupakan sinyal deret ke- n . Kelemahan yang dimiliki oleh DFT yaitu perhitungan yang banyak akibat dari perkalian berulang hingga N data. Akibat dari perhitungan yang banyak, transformasi sinyal berjalan dengan lambat. DFT tidak banyak digunakan dikarenakan efisiensi waktu yang kurang baik. Pada tahun 1965, Cooley dan Tukey menemukan algoritma DFT yang dapat disederhanakan. Algoritma yang ditemukan oleh Cooley dan Tukey memecah perhitungan yang

dilakukan oleh DFT sejumlah N data menjadi dua pecahan terkecil yang dinamakan *radix-2 decimation in time*. Dari Gambar 2.9 dan Persamaan 2.4 dapat dijabarkan menurut urutan antara data genap dan data ganjil. Sehingga membentuk Persamaan 2.6.

$$x[k] = \sum_{n=0}^{\frac{N}{2}-1} x[N_{\text{even}}] W_{N/2}^{nk/2} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[N_{\text{odd}}] W_{N/2}^{nk/2} \quad (2.6)$$

Dari Persamaan 2.5 dan Persamaan 2.6 didapatkan perbandingan jumlah perkalian antara DFT dan FFT dalam Tabel 2.2



Gambar 2.9 Diagram *Butterfly* FFT

Tabel 2.2 Perbandingan Jumlah Perhitungan yang Dilakukan Oleh DFT dan FFT

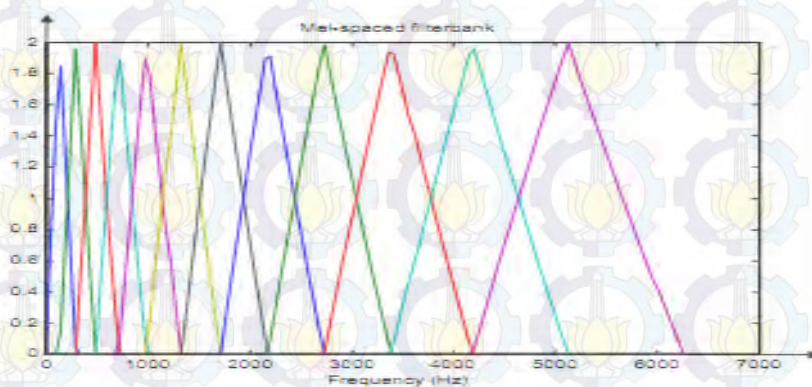
N	DFT	FFT
2	2	2
4	8	8
8	56	24
16	$16^2 - 16$	64
N	$N^2 - N$	$N \log_2 N$

d. *Mel-Frequency Wrapping*

Studi psikofisikal menunjukkan bahwa persepsi manusia dari kandungan frekuensi suara pada sinyal bicara tidak mengikuti skala linier. Untuk masingmasing nada dengan frekuensi aktual, f dalam Hz, *pitch* diukur dengan skala '*mel*'. Skala *mel-frequency* adalah frekuensi linier berada dibawah 1000 Hz dan bentuk logaritmik berada diatas 1000 Hz. Sebagai ti-tik referensi adalah *pitch* dengan *tone* 1 kHz, 40 d B diatas nilai batas ambang pendengaran, ini dinyatakan 1000 *mel*. Pendekatan persamaan untuk menghitung *mel* dalam *frekuensi f (Hz)* menggunakan Persamaan 2.7.

$$mel(f) = 2595 * 10 \log \left(1 + \frac{f}{700} \right) \quad (2.7)$$

Salah satu pendekatan simulasi spektrum yaitu menggunakan *filter bank*. *Filterbank* menggunakan representasi konvolusi dalam melakukan *filter* terhadap *signal*. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum *signal* dengan koefisien *filterbank*. Satu *filter* untuk masingmasing komponen *mel-frequency* yang diinginkan. *Filter bank* mempunyai respon frekuensi *bandpass* segitiga dan jarak *bandwidth* ditentukan oleh konstanta *interval mel-frequency*. Sebagai contoh filter bank dengan panjang 6000 Hz ditunjukkan pada Gambar 2.10.



Gambar 2.10 *Mel-Frequency Wrapping*

e. *Cepstrum*

Langkah selanjutnya yaitu mengubah spektrum *log mel* menjadi domain waktu. Hasil ini disebut *mel frequency cepstrum coefficient* (MFCC). Reprerentasi cepstral dari spectrum bicara memberikan reprerentasi baik dari sifat-sifat spektral lokal sinyal untuk analisis *frame* yang diketahui. Karena koefisien *mel* spektrum adalah bilangan nyata. Dengan mengubahnya menjadi domain waktu menggunakan *discrete cosine transform* (DCT). Kita dapat menghitung MFCC, seperti dalam Persamaan 2.8.

$$\tilde{C}_n = \sum_{k=1}^K (\log \tilde{S}_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad n = 1, 2, \dots, K \quad (2.8)$$

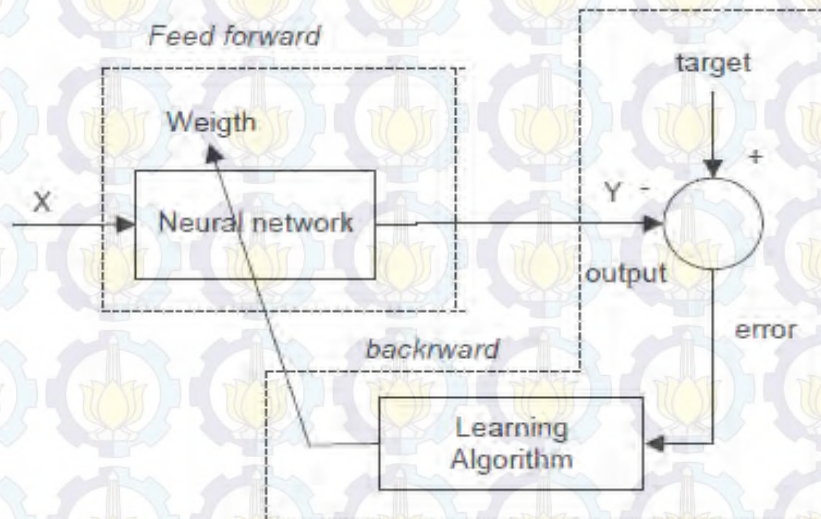
Dimana \tilde{C}_n adalah koefisien cepstrum *mel-frequency* dan \tilde{S}_k adalah koefisien daya *mel*.

2.4.3 Neural Network

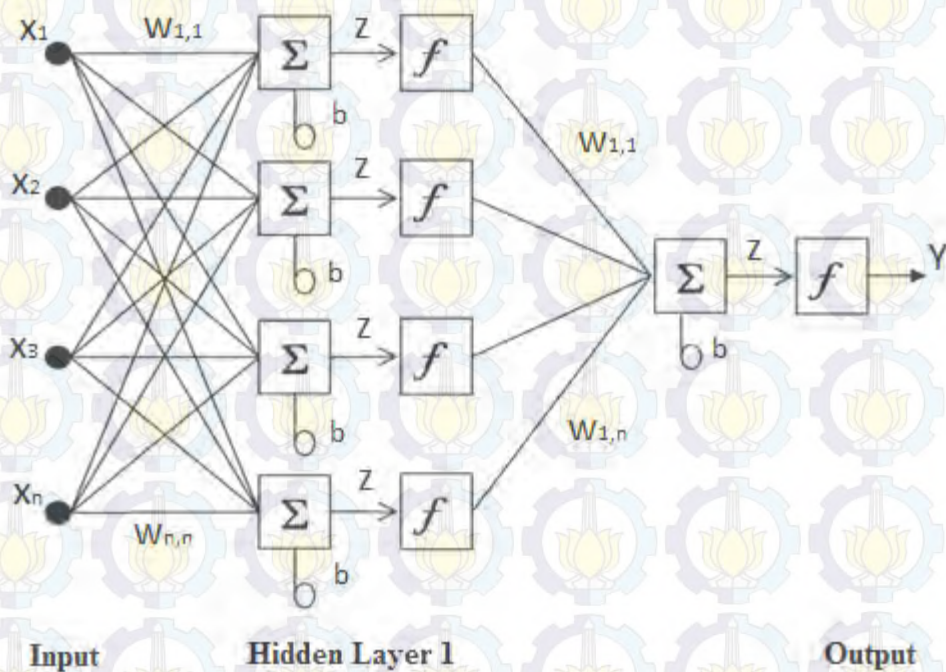
Neural network terdiri dari elemen-elemen sederhana yang meniru sistem saraf biologis manusia. *Neural network* dapat dilatih untuk melakukan fungsi tertentu dengan cara mengatur nilai-nilai dari koneksi (bobot) antar elemen. Secara umum, *neural network* dilatih agar *input* tertentu menghasilkan *output* target tertentu. Pada Gambar 2.11, jaringan dilatih berdasarkan perbandingan *output* dan targetnya sampai output jaringan sesuai dengan target. Proses pelatihan biasanya menggunakan banyak pasangan *input* dan *output* target yang diperlukan untuk melatih jaringan (Demuth dkk, 2009). Jaringan *neural network* memiliki koneksi satu arah dari input ke lapisan output. Gambar 2.12, menunjukan struktur dasar pemodelan *neural network* yang terdiri dari input dikalikan dengan masing-masing *wight*, selanjutnya dijumlahkan serta ditambahkan bias.

Keluaran dilakukan dengan fungsi pengaktif untuk mendapatkan derajat sinyal keluaran. Beberapa algoritma pelatihan *neural network* diantaranya adalah *backpropagation*, *learning vector quantization*, *radial base fuction*, *hopfield and kohonen*. *Backpropagation* adalah salah satu metode pembelajaran yang dapat

diterapkan dalam *Neural Network* (NN). Arsitektur jaringannya tidak memiliki koneksi umpan balik tetapi *error* dipropagasikan selama dilatih.



Gambar 2.11 Neural Network (Demuth dkk, 2009)



Gambar 2.12 Struktur Dasar Pemodelan *Neural Network*

Pada pembelajaran ini digunakan *error* dengan perhitungan *Mean Square Error* (MSE). Algoritma pelatihan *Backpropagation Neural Network* terdiri dari dua tahap, yaitu *feed forward* dan *backward propagation*.

a. Tahap *feed forward*

Pada Gambar 2.11 merupakan proses *feed forward*, pertama memberikan *X input* dan targetnya. Kemudian melakukan setting bobot (*weight*) secara random selanjutnya dilakukan perhitungan untuk mendapatkan nilai *Y output*.

Masing-masing unit lapisan dikalikan *weight* dan dijumlahkan serta ditambahkan dengan bias:

$$z = b + \sum_{i=1}^n X_i W_{ij} \quad (2.9)$$

Dihitung dengan fungsi aktivasi:

$$Y = f(z) \quad (2.10)$$

Kemudian dihitung pada fungsi sigmoid yang bentuk fungsinya adalah:

$$f(z) = \frac{1}{1 + \exp(-z)} \quad (2.11)$$

Sinyal keluaran dari fungsi aktivasi tersebut dikirimkan ke semua unit dilapisan luar. Tiap-tiap output layer dikalikan dengan *weight* dan dijumlahkan dengan bias serta dihitung lagi fungsi aktifasinya. Proses yang sama dilakukan hingga layer output.

b. Tahap *Backward*

Tahap *backward* merupakan proses perhitungan *error* dan mengatur ulang bobot tiap-tiap layer, setelah diperoleh nilai output pada proses *feedforward* (*Y*) maka akan dilakukan proses perhitungan error untuk kemudian nilai error tersebut

akan digunakan sebagai variabel untuk mengupdate bobot tiap-tiap layer. Jika ada selisih antara output yang diharapkan dengan output sebenarnya (error) maka akan disebarkan mundur (*backward*) pada tiap layer-nya, untuk lebih jelasnya dapat dilihat pada Gambar 2.11. Oleh karena itu proses tersebut disebut sebagai propagasi balik (*backpropagation*) dimana tahap pelatihan dilakukan dengan merubah bobot yang menghubungkan unit dalam layer *neural network*. Masing-masing *output layer* menerima pola target sesuai dengan pola masukan saat *training* dan dihitung *error*-nya.

$$\Delta = (Y_{target} - Y)f'(z) \quad (2.12)$$

$$f'(z) = f(z)(1 - f(z)) = Y(1 - Y) \quad (2.13)$$

Delta merupakan selisih target yang ingin dicapai dengan nilai keluaran saat ini dikalikan dengan turunan dari fungsi aktivasi yang digunakan. Delta dapat dihitung menggunakan Persamaan 2.12. Sedangkan turunan fungsi aktivasi dapat dicari dengan menggunakan Persamaan 2.13. Masing-masing output layer diperbaiki bias dan *weight*-nya. Nilai pembaruan pembobot dan *bias* dapat dihitung dengan Persamaan 2.14 dan Persamaan 2.15.

$$W_{1,n}(baru) = W_{1,n}(lama) + \Delta\mu x_i \quad (2.14)$$

Seluruh *hidden layer* diperbaiki *bias* dan *weight*.

$$b_{1,n}(baru) = b_{1,n}(lama) + \Delta\mu \quad (2.15)$$

Pada akhirnya, nilai pembaharuan *weight* dan *bias* ini yang dijadikan acuan pada proses pengujian. Nilai ini ibarat memori yang dapat digunakan dalam proses pengenalan pola. Proses pengujian itu sendiri merupakan proses penggolongan pola baru berdasarkan sampel pola yang telah dipelajari sebelumnya.

2.5 Pengolahan Citra

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra atau gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi kontinu (continue) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reperesentasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra. Normalisasi RGB adalah salah satu metode segmentasi warna yang memiliki kelebihan yaitu mudah, proses cepat dan efektif pada obyek *traffic sign*, maupun aplikasi untuk *face detection*. Segmentasi warna, ada bermacam-macam model warna. Model RGB (*Red Green Blue*) merupakan model yang banyak digunakan, salah satunya adalah monitor. Pada model ini untuk merepresentasikan gambar menggunakan tiga buah komponen warna tersebut. Selain model RGB terdapat juga model normalisasi RGB dimana model ini terdapat 3 komponen yaitu, r , g , b yang merepresentasikan prosentase dari sebuah piksel pada citra digital. Nilai-nilai tersebut mengikuti Persamaan 2.16 dan Persamaan 2.17.

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \quad (2.16)$$

$$r + g + b = 1 \quad (2.17)$$

Dengan demikian berdasarkan persamaan 2.8 maka cukup hanya menggunakan r dan g saja, karena nilai b bisa didapatkan dengan menggunakan $b = 1 - r - g$. Salah satu platform yang digunakan dalam proses pengolahan citra yaitu EmguCV. EmguCV adalah cross platform yang terdapat dalam .NET untuk *library* pengolahan citra pada Intel OpenCV. EmguCV ini mengikuti fungsi yang terdapat pada OpenCV yang diambil dari .NET oleh sebab itu *compatible* dengan bahasa pemrograman C#, VB, VC++, IronPython dan sebagainya. Program ini

bersifat *opensource* sehingga sangat cocok apabila digunakan untuk penelitian, salah satunya adalah untuk aplikasi *computer vision*. Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi tiga, *color image*, *black and white image* dan *binary image*.

2.5.1. Color Image atau RGB (Red, Green, Blue).

Pada *color image* ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika masing-masing warna memiliki range 0-255, maka totalnya adalah $255^3 = 16.581.375$ variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap pixel, gambar tersebut juga disebut gambar-bit warna. *Color image* ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan Gambar 2.13.



Gambar 2.13 Color Image (Alasdair, 2004)

2.5.2 Black and White

Citra digital *black and white* (*grayscale*) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 byte. Rentang warna pada *black and white* sangat cocok digunakan untuk pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam kedokteran (X-ray). *Black and white* sebenarnya merupakan hasil rata-rata dari *color image*, dengan demikian maka persamaannya dapat dituliskan pada Persamaan 2.18.

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.18)$$

Dimana $I_R(x, y)$ merupakan nilai piksel *Red* titik (x, y) , $I_G(x, y)$ nilai piksel *Green* titik (x, y) , $I_B(x, y)$ nilai piksel *Blue* titik (x, y) sedangkan $I_{BW}(x, y)$ adalah nilai piksel *black and white* titik (x, y) . Gambar 2.14 merupakan contoh gambar hasil dari *Black and white*.



Gambar 2.14 *Black and White* (*Grayscale*) (Alasdair, 2004)

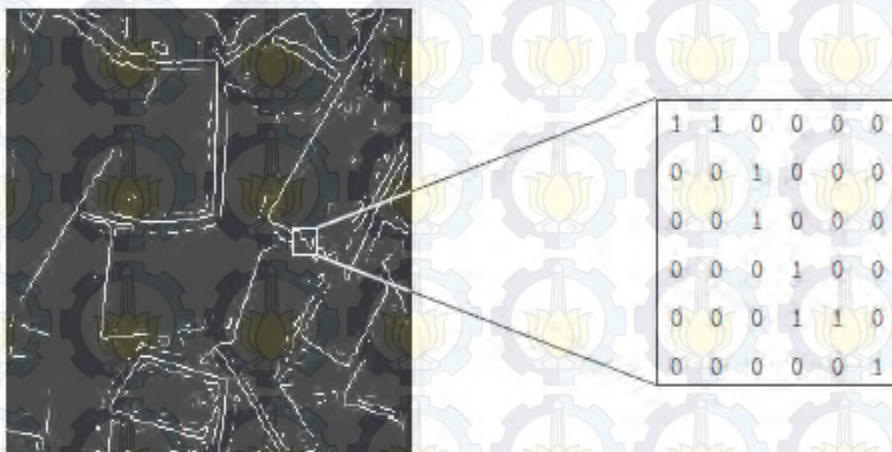
2.5.3 Binary Image

Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (*finger print*), atau gambar arsitektur. *Binary image* merupakan hasil pengolahan dari *black and white image*, dengan menggunakan fungsi pada Persamaan 2.19 dan dalam bentuk *floating point* menggunakan Persamaan 2.20.

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 255 & I_{BW}(x,y) \geq T \end{cases} \quad (2.19)$$

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 1 & I_{BW}(x,y) \geq T \end{cases} \quad (2.20)$$

Dimana $I_{BW}(x,y)$ merupakan nilai piksel *Gray* titik (x,y) , $I_{Bin}(x,y)$ nilai piksel *Binary* titik (x,y) , sedangkan T adalah nilai *threshold*. Contoh dari hasil *binary image* ditunjukkan pada Gambar 2.15.



Gambar 2.15 Image Binary (Alasdair, 2004)

2.6 Robot Lengan

Sebuah lengan robot adalah jenis lengan mekanik, biasanya diprogram, dengan fungsi yang mirip dengan lengan manusia. Terdiri dari beberapa link yang disebut dengan derajat kebebasan. Link dari manipulator tersebut dihubungkan oleh sendi yang memungkinkan baik gerak rotasi (seperti dalam sebuah robot diartikulasikan) atau translasi (linear) perpindahan. Link dari manipulator dapat dianggap untuk membentuk rantai kinematik. Ujung rantai kinematik manipulator disebut *end effector* dan merupakan analogi dengan tangan manusia. Robot lengan terdiri dari empat komponen dasar yaitu manipulator, kontroler, *power supply* dan *end effector*. Robot mempunyai bermacam-macam bentuk dan ukuran, sehingga memiliki beragam kemampuan gerakan. Secara fisik, terdapat beberapa konfigurasi yang dapat dibentuk.

a. Konfigurasi Koordinat Kartesian

Robot kartesian memiliki tiga sendi linear yang menggunakan sistem koordinat kartesian (X, Y, dan Z). Robot juga mungkin memiliki pergelangan tangan yang melekat untuk memungkinkan gerakan rotasi. Tiga sendi prismatic memberikan gerakan linier sepanjang sumbu kartesian.

b. Konfigurasi Koordinat Silinder

Sistem koordinat silinder memiliki tiga derajat kebebasan (DOF) atau tiga axis, yang terdiri dari θ (theta) mewakili sumbu putar, sumbu z mewakili gerakan naik-turun serta sumbu R yang mewakili gerakan memanjang atau memendek.

c. Konfigurasi Koordinat Polar

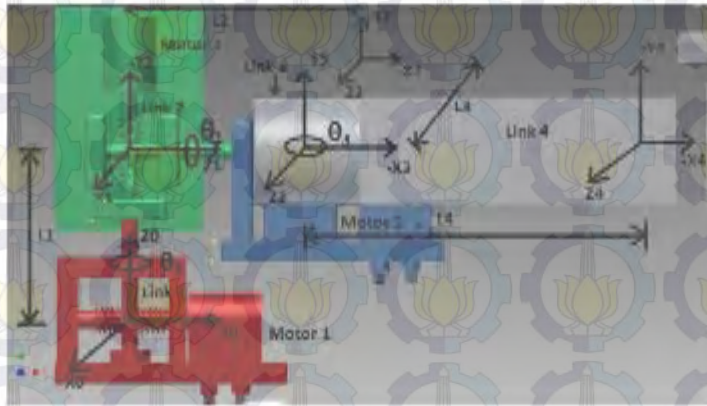
Konfigurasi koordinat polar/simetrikal juga memiliki tiga sumbu yaitu θ (theta), β (beta), dan R. dikatakan sistem simetrikal karena ruang gerak dari robot merupakan sphere (bola).

d. Sistem Koordinat Articulate

Sistem koordinat articulate didefinisikan dengan tiga sumbu, yakni θ (theta), upper arm (w) dan elbow (U). Sumbu ini memberikan ke-fleksibelan lebih besar.

e. SCARA (*Selective Compliance Assembly Robot Arm*)

Sistem sumbu yang mirip koordinat Articulate tetapi berbasis pada gerakan horizontal. SCARA memiliki kemampuan untuk “insektion”.



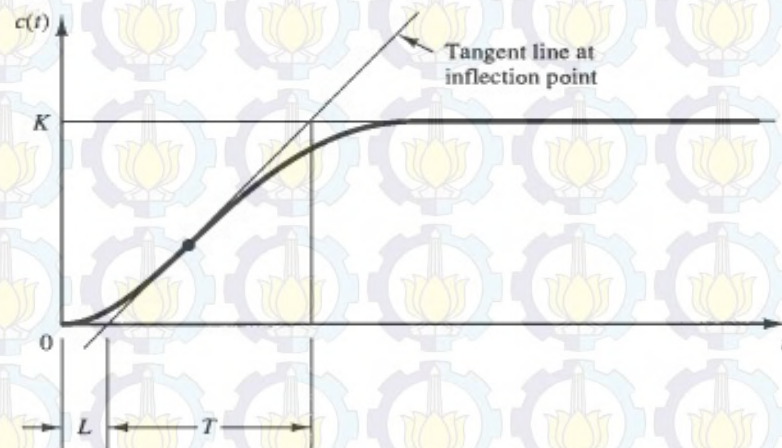
Gambar 2.16 Model Arm Robot 3 Dof Parekh

Berdasarkan konfigurasi gerak pada robot lengan parekh dkk membuat model robot seperti pada Gambar 2.16. Model robot lengan yang dibuat oleh parekh terdiri dari tiga derajat kebebasan. Derajat kebebasan pertama terletak bagian bawah berupa gerak rotasi yaitu, gerak berputar 360 derajat. Derajat kebebasan kedua berupa gerak translasi yaitu, robot lengan dapat bergerak ke atas dan kebawah. Derajat kebebasan yang ketiga juga berupa gerak translasi yaitu, memungkinkan robot dapat menggerakkan lengannya ke depan dan ke belakang.

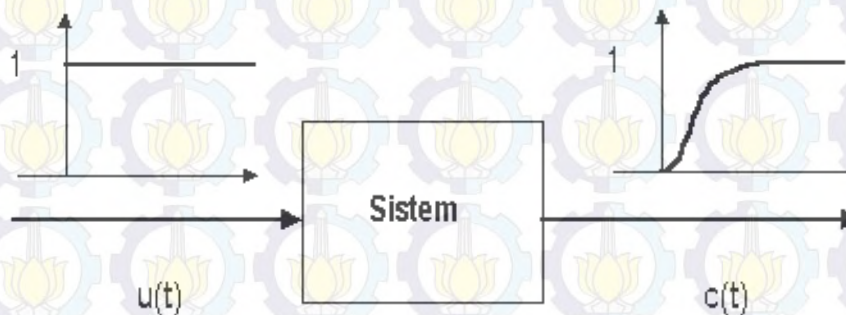
2.7 Ziegler Nichols Untuk Tuning PID

Pemilihan kontrol PID dan variasinya (P, PI, PD) dan parameter-parameternya (K_p , K_i , dan K_d) erat hubungannya dengan model dari proses yang dikontrol. Pemilihan dari parameter-parameter kontrol pada dasarnya adalah optimasi masalah dimana memenuhi beberapa performansi kriteria sistem kontrol yang dibuat. Pada sistem kontrol PID jika terjadi gangguan pada proses melewati batas linieritas PID, maka diperlukan tuning ulang agar kontroler dapat bekerja secara optimal. Salah satu metode untuk memudahkan tuning adalah metode Ziegler Nichols yang diperkenalkan oleh John G. Ziegler dan Nathaniel B. Nichols. Ziegler-Nichols terdapat beberapa metode untuk menentukan nilai K_p , K_i , K_d untuk kontroler PID, yang berdasarkan respon langkah transien terhadap plant. Dalam metode ini, pada Gambar 2.16 kurva reaksi ditentukan oleh dua parameter L (*delay time*) dan T (waktu konstan) (Melani dkk, 2012). Pada metode

ini yang pertama dilakukan adalah memberikan input step pada sistem dengan kondisi open loop, maka reaksi sistem akan berbentuk S sebagaimana yang ditunjukkan pada Gambar 2.17. Setelah mendapatkan kurva reaksi berbentuk S, maka langkah selanjutnya adalah mencari gradien terbesar pada titik-titik sepanjang kurva S tersebut yang kemudian disebut titik infleksi sebagaimana yang ditunjukkan pada Gambar 2.18. Dari titik infleksi tersebut ditarik garis yang menyinggung banyak dua konstanta yaitu konstanta L dan T. Konstanta L dan T ini akan digunakan untuk tuning PID sebagaimana yang ditunjukkan pada Tabel 2.3.



Gambar 2.17 Kurva Reaksi



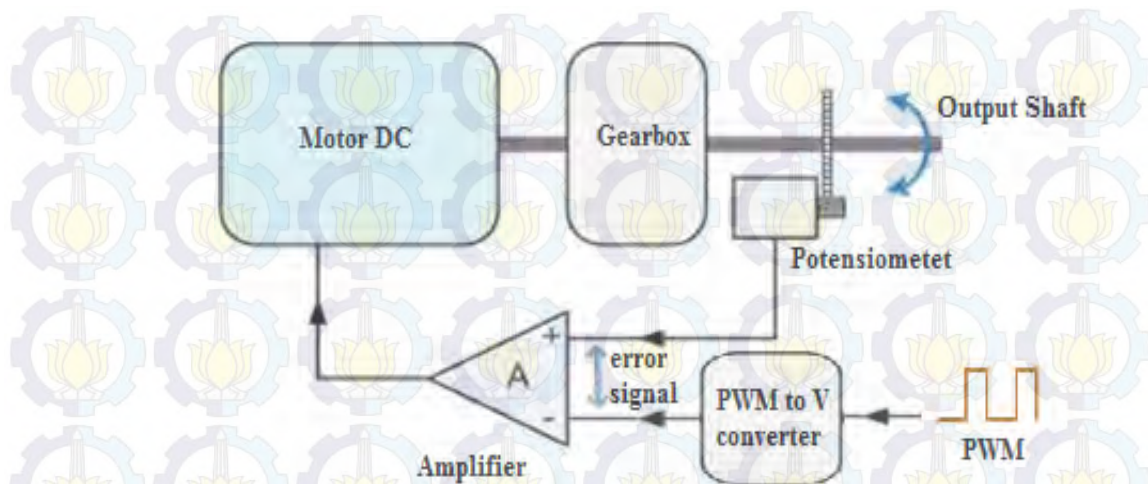
Gambar 2.18 Reaksi sistem open loop ketika diberi input step

Tabel 2.3 Tuning PID metode Ziegler-Nichols

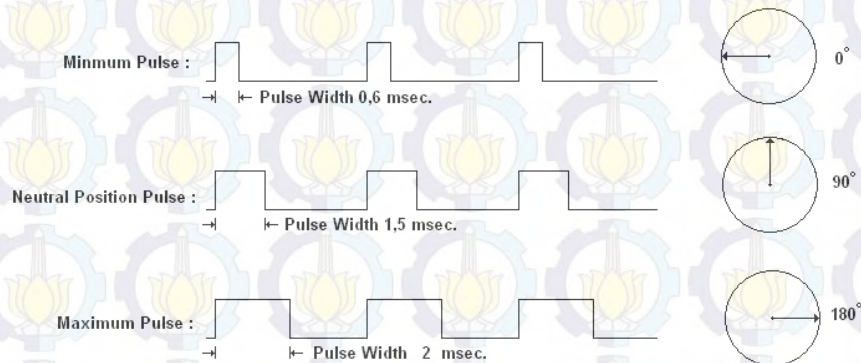
Metode Tuning Ziegler-Nichole			
Tipe Kontrol	Kp	Ti=Kp/Ki	Td=Kd/Kp
P	T/L	-	-
PI	0.9T/L	L/0.3	-
PID	1.2T/L	2L	0.5L

2.8 Motor Servo

Motor servo adalah sebuah alat yang dapat mengubah energi listrik menjadi energi mekanik. Motor servo sebenarnya digerakkan oleh sebuah motor DC. Karena motor servo digerakkan oleh motor DC maka jenis arus yang dipergunakan adalah jenis arus searah. Selain terdapat komponen motor DC, di dalam motor servo pun terdapat roda gigi yang berfungsi untuk mereduksi putaran dan untuk memperbesar torsi yang dihasilkan oleh motor DC tersebut. Motor servo juga memiliki controller dan potensiometer didalamnya. Motor DC dan potensiometer dihubungkan ke controller, sehingga ketiganya membentuk sistem close loop, seperti pada Gambar 2.19. Serangkaian gear pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Tegangan yang dihasilkan potensiometer akan dibandingkan dengan tegangan input. Jika tegangan yang dihasilkan potensiometer sama dengan tegangan input maka error yang dihasilkan sama dengan nol. Error yang berupa tegangan di feedbackkan kembali ke motor DC sehingga membuat motor tidak bergerak atau diam. Hal ini menunjukan bahwa output dari motor servo sudah berada pada posisi sesuai dengan yang diinputkan. Salah satu jenis motor servo adalah motor servo jenis standar. Motor servo jenis standar hanya dapat berputar 180° searah atau berlawanan arah jarum jam. Prinsip kerja motor servo standar adalah kaki sinyal motor servo diberi sinyal digital dengan lebar sinyal antara 0,60 milidetik sampai 2,00 milidetik. Bentuk sinyal pengontrolan motor servo dapat dilihat pada Gambar 2.20.

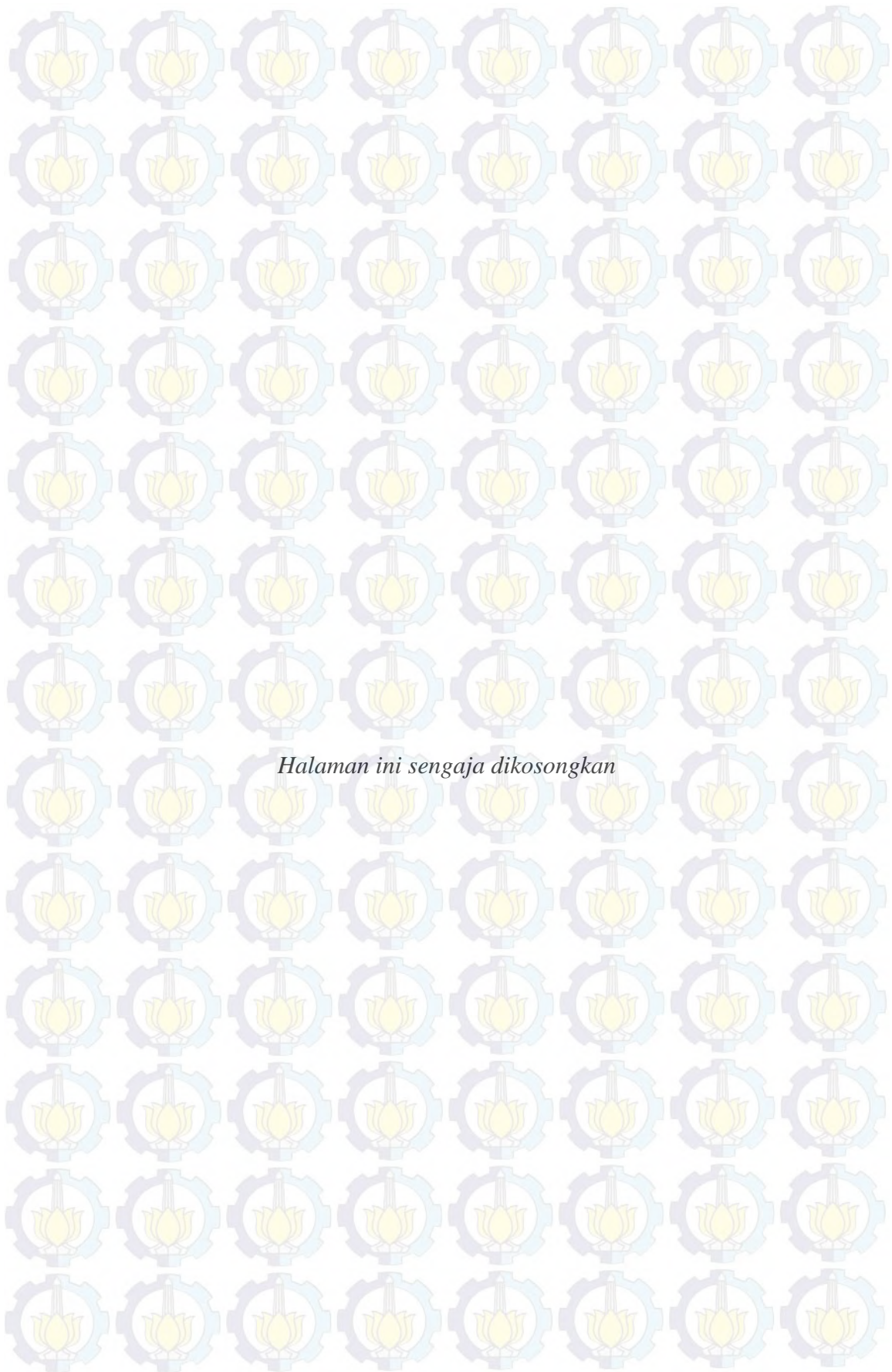


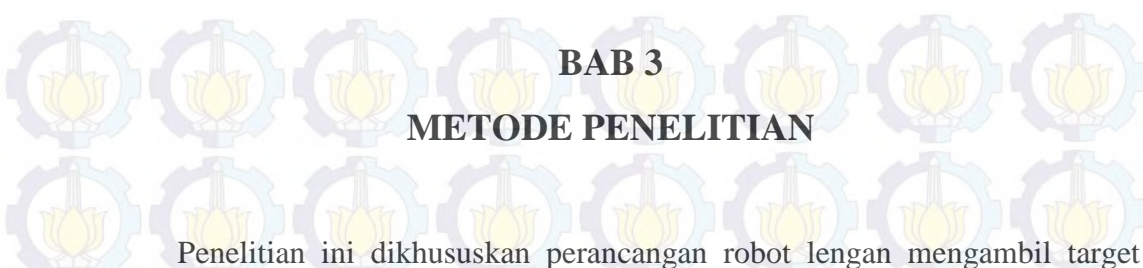
Gambar 2.19 Sistem Close Loop Motor Servo



Gambar 2.20 Bentuk sinyal pengontrolan motor servo

Untuk memposisikan poros motor servo pada sudut tertentu dan selama waktu tertentu, lebar dan jumlah pulsa tertentu harus diberikan pada kaki sinyal motor servo. Agar motor servo dapat diposisikan pada sudut tertentu terlebih dahulu lebar pulsa untuk posisi 0° dan posisi 180° harus diketahui. Setelah mengetahui lebar pulsa pada kedua posisi tersebut, lebar pulsa untuk posisi sudut yang lain bisa diketahui dengan cara interpolasi. Karena lamanya pulsa yang diberikan sangat singkat maka pemberian pulsa pada kaki sinyal tidak mungkin dilakukan secara manual. Selain itu sejumlah pulsa harus diberikan selama selang waktu tertentu. Oleh karena itu, pemberian pulsa pada kaki sinyal motor servo dilakukan dengan menggunakan mikrokontroler.



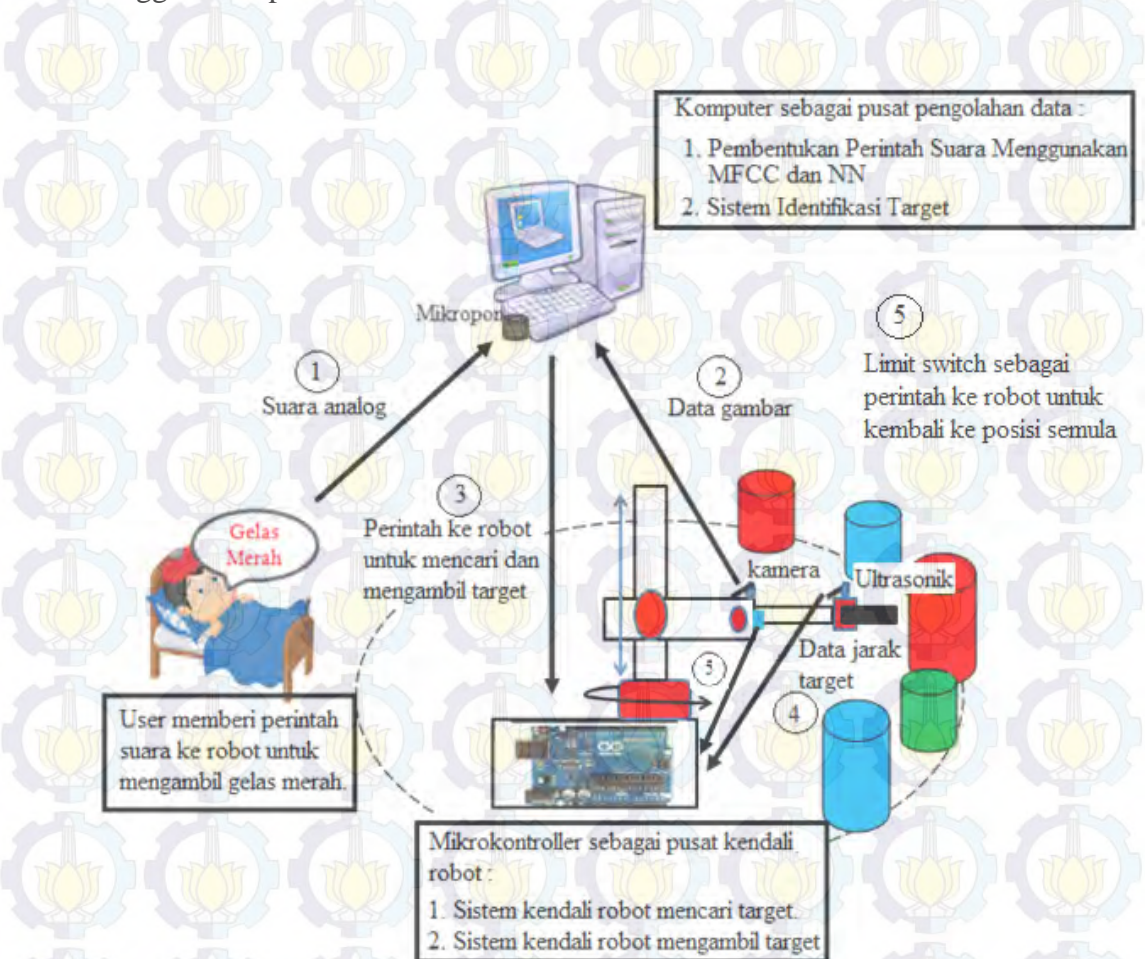


BAB 3

METODE PENELITIAN

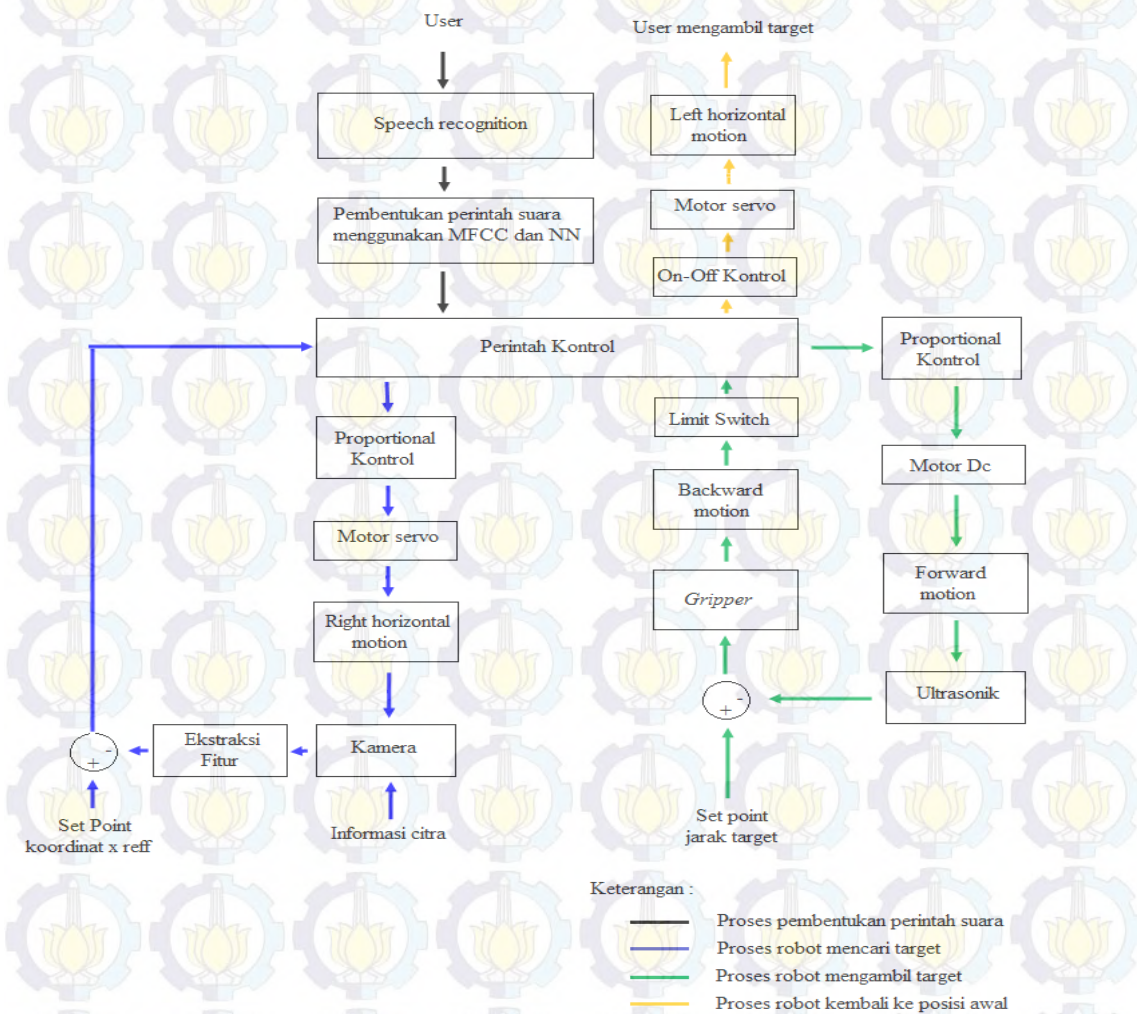
Penelitian ini dikhususkan perancangan robot lengan mengambil target

dengan menggunakan perintah suara. Robot ini pada peancangannya digunakan sebagai robot pelayan untuk membantu pasien dalam mengambil target yang diinginkan. Pasien yang dimaksud disini memiliki keterbatasan gerak sehingga tidak dapat mengambil target sendiri. Target yang digunakan berupa gelas dan botol yang memiliki warna berbeda-beda. Letak target diletakan secara acak dengan jarak target tidak lebih dari panjang maksimum lengan robot. Gambar 3.1 merupakan ilustrasi cara kerja dari robot lengan mengambil target dengan menggunakan perintah suara.



Gambar 3.1 Ilustrasi Cara Kerja Robot Lengan Pengambil Target Menggunakan Perintah Suara

Pasien atau subjek memerintahkan robot untuk mengambil target dengan cara menyebutkan nama target yang diinginkan. Sebagai contoh pada Gambar 3.1 pasien memerintahkan robot lengan untuk mengambil gelas merah. Perintah suara berupa suara analog "gelas merah" direkam oleh komputer dengan menggunakan mikropon. Komputer digunakan sebagai pusat pengolahan data pembentukan perintah suara dan identifikasi target. Data keluaran dari komputer dikirim ke mikrokontroler berupa sebuah perintah untuk menggerakkan robot. Mikrokontroler digunakan sebagai pusat kendali untuk gerak robot dalam mencari dan mengambil target yang diperintahkan. Proses robot lengan mengambil target menggunakan perintah suara ditunjukkan pada blok diagram Gambar 3.2.



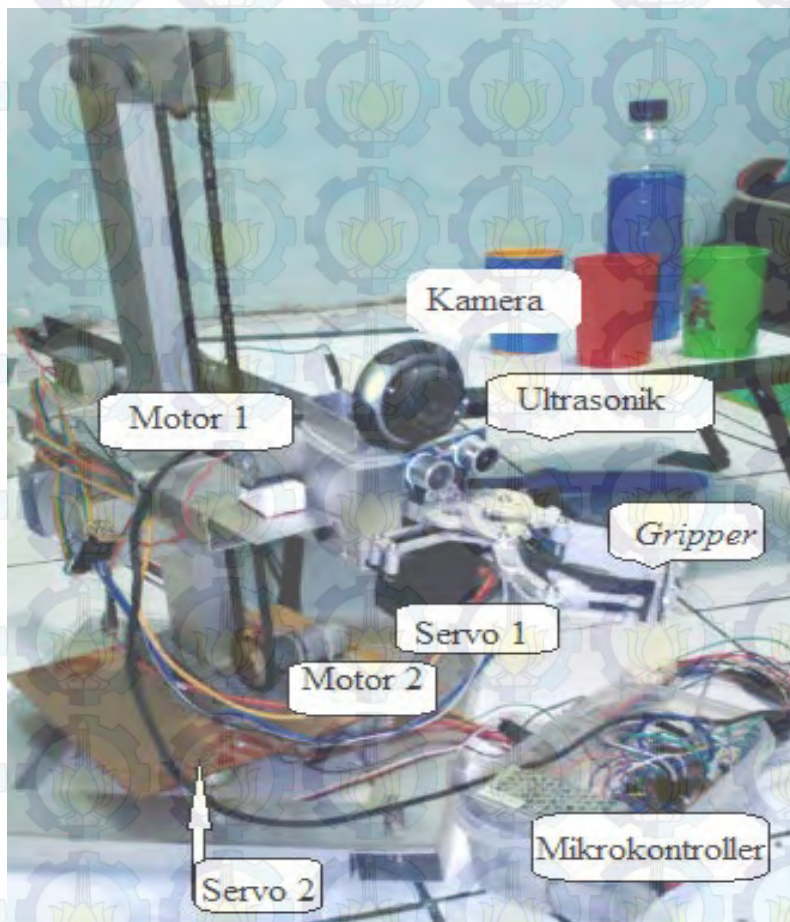
Gambar 3.2 Blok Diagram Proses Cara Kerja Robot Lengan Pengambil Target Menggunakan Perintah Suara

Tahap pertama yaitu proses pembentukan perintah suara. Pasien sebagai user memberikan perintah suara ke sistem. Suara dari user direkam untuk pembentukan sebuah perintah suara. Metode perintah suara yang digunakan yaitu pengenalan pola MFCC. Sebagai pengambil keputusan dari pola perintah suara yang dibentuk menggunakan neural network. Setelah perintah suara dikenali sistem akan mengidentifikasi target berdasarkan perintah yang diberikan. Data gambar diambil menggunakan kamera sebagai identifikasi dari target. Setelah perintah suara dan identifikasi target dikenali, proses kedua yaitu robot mencari target. Sistem memberikan perintah ke robot untuk mencari target. Proportional kontrol digunakan dalam proses robot mencari target. Robot menggerakkan motor servo berputar ke kanan untuk mencari target. Kamera diletakan pada lengan robot untuk mencari target. Set point yang digunakan berupa nilai tengah horizontal (x_{ref}) dari frame yang digunakan. Error didapat dari selisih set point terhadap nilai titik tengah horizontal dari target yang diambil kamera. Setelah error yang didapat sama atau mendekati nol, proses ketiga yaitu robot mengambil target. Sistem akan memerintahkan robot untuk mengambil target. Proportional kontrol juga digunakan dalam proses robot mengambil target. Robot menggerakkan motor dc untuk maju memanjangkan lengan mendekati target. Ultrasonik digunakan untuk mengetahui jarak lengan terhadap target. Set point yang digunakan berupa jarak ujung lengan robot terhadap target. Error didapat dari selisih set point terhadap pembacaan nilai jarak target oleh ultrasonik. Setelah error yang didapat sama atau mendekati nol, gripper akan menggengam target. Proses terakhir yaitu robot membawa target kembali ke posisi awal robot mulai yaitu posisi dimana user berada.

3.1 Spesifikasi Robot Lengan

Robot lengan yang digunakan terbuat dari bahan almunium dengan ukuran tinggi 48cm dan lebar 60cm. Robot lengan memiliki tiga *degree of freedom* dalam gerakannya. Pergerakan robot terdiri dari satu gerak rotasi dan dua gerak translasi. Gerak rotasi digunakan pada robot dalam mencari target. Sedangkan gerak translasi digunakan pada robot untuk mengambil target dan untuk naik-turunkan lengan robot. Motor servo *countinuous* digunakan pada gerak rotasi robot.

Sedangkan untuk gerak translasi menggunakan motor DC 12V. Ujung lengan robot dilengkapi dengan *gripper* dan sensor ultrasonik. Gripper yang digunakan memiliki spesifikasi panjang 12cm, lebar 10cm ketika menutup dan 20cm ketika terbuka. Spesifikasi dari robot lengan yang digunakan secara lengkap ditunjukkan pada Gambar 3.3. *Gripper* sebagai *end-effector* dari lengan robot digunakan untuk menggenggam target yang ingin diambil. *Gripper* juga dilengkapi dengan *limit switch* pada salah satu ujungnya. *Limit switch* pada *gripper* digunakan sebagai tanda bahwa *gripper* sudah menggenggam target. Motor servo digunakan untuk membuka dan menutup *gripper*. Sensor ultrasonik digunakan untuk mengetahui dan memandu lengan robot terhadap jarak target. Lengan robot juga dilengkapi dengan kamera. Kamera yang digunakan merupakan kamera jenis *webcam* dengan resolusi sebesar 16 megapiksel. Kamera digunakan pada robot untuk mengenali target yang akan diambil.

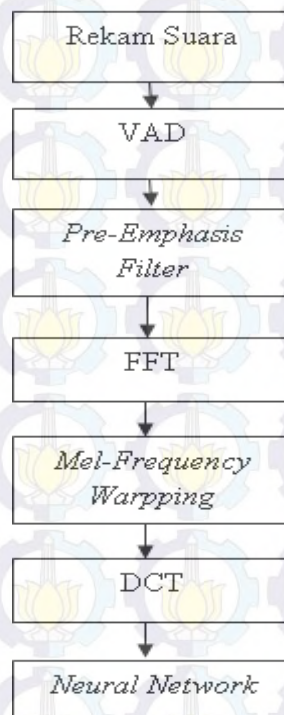


Gambar 3.3 Spesifikasi Robot Lengan

Kamera akan membedakan mana objek yang akan menjadi target dan bukan target. Pusat kontrol untuk robot lengan menggunakan arduino mega 2560 R3 dengan tegangan input sebesar 5V. Sedangkan driver untuk motor dc dan motor servo menggunakan arduino motor shield V1.2

3.2 Pembentukan Perintah Suara Menggunakan MFCC dan NN

Robot lengan diperintah dengan menggunakan perintah suara dengan cara menyebutkan nama target yang ingin diambil. Metode pembentukan perintah suara yang digunakan adalah *Mel-Frequency Cepstrum Coefficients* (MFCC) dan *Neural Network*. MFCC merupakan suatu metode yang cara bekerjanya seperti telinga manusia. MFCC mampu merubah suara analog dan membentuknya menjadi suatu pola yang dapat dikenali. Pola-pola dari setiap kata digunakan sebagai input neural network sebagai data pembelajaran. Keluaran dari neural network merupakan kode yang direpresentasikan sebagai perintah suara. Gambar 3.4 merupakan tahapan dari pembentukan perintah suara menggunakan MFCC dan *neural network*.



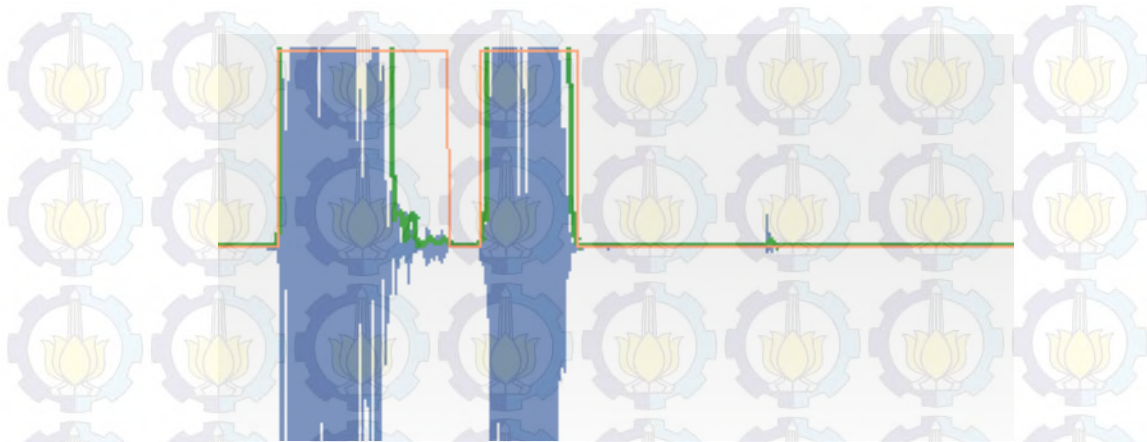
Gambar 3.4 Blok Diagram Pembentukan Perintah Suara

Proses awal pembentukan perintah suara yaitu merekam suara dari kata yang digunakan sebagai perintah suara. Suara direkam menggunakan mikropon yang terhubung dengan komputer. Pada penelitian ini kata yang digunakan adalah "gelas", "botol", "merah", "biru", dan "hijau". Kata-kata tersebut direkam dan diproses untuk mengetahui pola atau ciri dari setiap kata. Setelah perekaman kata yang digunakan, langkah selanjutnya adalah *Voice Activity Detection* (VAD). Suara terdiri dari suara sebagai informasi dan suara sebagai noise. Dalam mendeteksi suara informasi, noise merupakan masalah kompleks dalam proses pengenalan suara. VAD atau *silence detection* merupakan metode untuk membedakan segmen suara informasi dan suara noise. Cara kerja dari VAD yaitu memberi batas dari power suara terekam sehingga dapat dibedakan mana suara sebagai informasi dan suara diam. Gambar 3.5 merupakan proses VAD dari kata "gelas merah". VAD pada Gambar 3.5 dipresentasikan oleh garis warna orange. Kata "gelas" dan kata "merah" merupakan suara sebagai informasi yang terdeteksi oleh VAD. Sedangkan yang tidak terdeteksi oleh VAD dianggap sebagai noise. Hasil dari VAD akan dipotong perkata untuk dianalisa bentuk polanya. Gambar 3.6 merupakan VAD dari kata "gelas" yang dipotong. *Pseudocode* untuk VAD yang digunakan sebagai berikut.

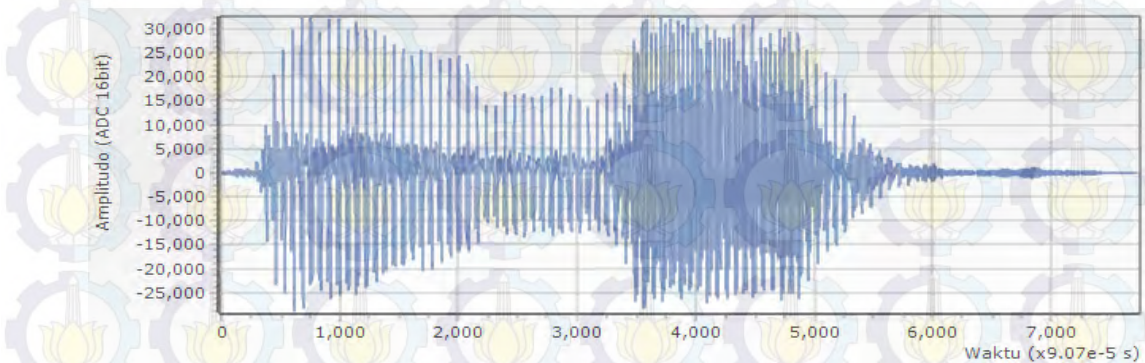
```

for i := 0 to 132300 do
begin
    sigPowSub[i]:=abs(sigPow[i]-sigPow[i-1]);
end;
for i := nFrame to 132300 do
begin
    temp:=0;
    for j := 1 to nFrame do
        temp:=temp+sigPowSub[i-j];
    temp:=temp/nFrame;
    if (temp<batas bawah) then
        sigVAD[i+nFrame]:=0
    else
        sigVAD[i+nFrame]:=batas atas;
    end;
end;

```

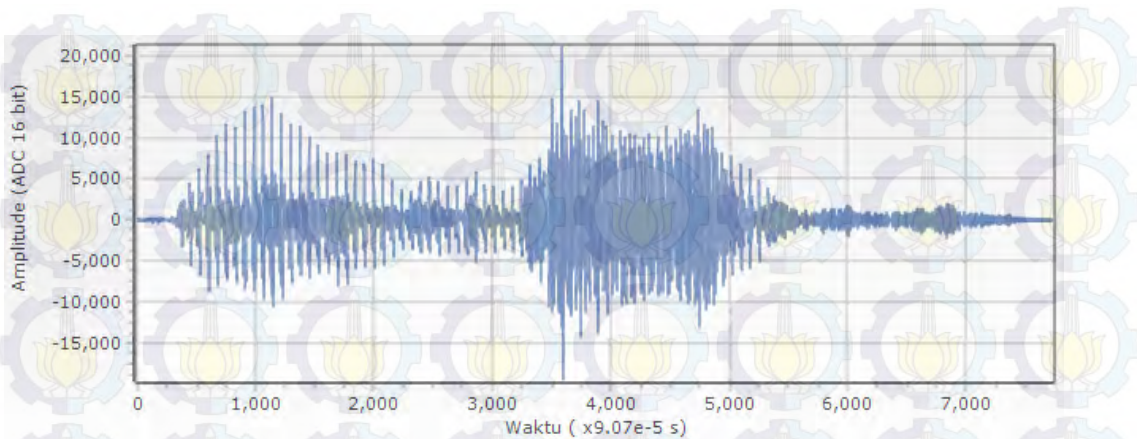
Gambar 3.5 VAD dari Kata "gelas merah"



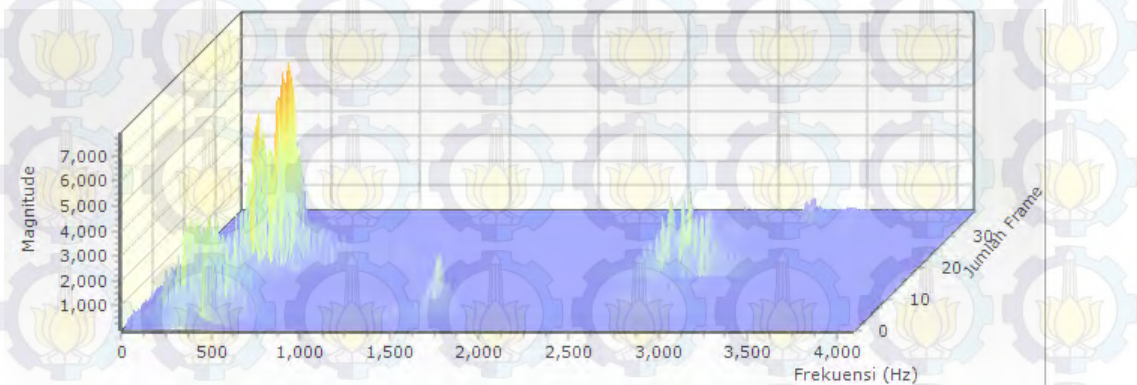
Gambar 3.6 VAD Kata "gelas"

Potongan suara dari proses VAD dilakukan *pre-emphasis filter*. *Pre-emphasis filter* akan mengkompensasi bagian frekuensi tinggi yang terekam. Selain itu *pre-emphasis filter* juga digunakan untuk menguatkan frekuensi tinggi. *Pre-emphasisfiltering* dibentuk dengan mengurangi sinyal dengan sinyal sebelumnya. Informasi sinyal frekuensi tinggi akan dapat diamati dengan menggunakan *pre-emphasis filter*. Gambar 3.7 merupakan *pre-emphasis filter* dari kata "gelas". *Pseudocode* untuk *pre-emphasis filter* dipresentasikan sebagai berikut.

```
for i := 0 to iStop-iStart do
    sigPEF[i]:=sigRaw [i]-alpha*sigRaw [i-1];
```

Gambar 3.7 Pre-Emphasis Filter Kata "gelas"



Gambar 3.8 FFT Kata "gelas"

Sinyal dalam domain waktu diubah ke dalam domain frekuensi dengan rentang 0–4 kHz. Analisa *fast fourier transform* dilakukan per-frame sehingga setiap frame-nya akan menghasilkan 4000 data *magnitude* frekuensi sinyal. Dari 4000 data untuk setiap frame akan dimampatkan menjadi 24 data untuk setiap frame. Gambar 3.8 merupakan *fast fourier transform* dari kata "gelas". Fungsi fft yang digunakan sebagai berikut.

```
procedure TfmMain.fft(t:extended);
label satu,dua,tiga,empat;
var
    lim1,lim2,lim3,l,r,m,k,i,j:integer;
    d,x1,x2,b1,b2,c1,c2,arg,sin1,cos1:extended;
begin
```



```

lim1:=panjang_fft-1;
lim2:= round(panjang_fft/2);
j:= 1;
for i:= 1 to lim1 do
begin
    if i > j-1 then
begin
        rec[i]:=j;
        goto dua;
    end
else
    rec[i]:=j;
    x1:=xr[j];
    x2:=xi[j];
    xr[j]:=xr[i];
    xi[j]:=xi[i];
    xr[i]:=x1;
    xi[i]:=x2;
    dua:
    l:=lim2;
    tiga:
    if l > j-1 then
        goto empat;
    j:=j-1;
    l:=trunc(l/2);
    goto tiga;
    empat:
    j:=j+1;
end;
{inplace transformation}
for i:= 1 to m do
begin
    lim1:= trunc(exp((i-1)*ln(2)));
    lim2:= trunc(exp((m-i)*ln(2)));
    for l:= 1 to lim2 do
    begin
        for r:= 1 to lim1 do

```



```

begin
    lim3:= (r-1)+(l-1)*2*lim1+1;
    b1:=xr[lim3];
    b2:=xi[lim3];
    c1:=xr[lim3+lim1];
    c2:=xi[lim3+lim1];
    arg:=2*pi*(r-1)*lim2/panjang_fft;
    cos1:=cos(arg);
    sin1:=sin(arg);
    x1:=c1*cos1+c2*sin1*t;
    x2:=-c1*sin1*t+c2*cos1;
    xr[lim3]:=b1+x1;
    xi[lim3]:=b2+x2;
    xr[lim3+lim1]:=b1-x1;
    xi[lim3+lim1]:=b2-x2;
end;
end;
end;
for k:= 1 to panjang_fft do
begin
    xr[k-1]:=xr[k]/d;
    xi[k-1]:=xi[k]/d;
end;
end;
end;

```

Setelah nilai frekuensi kata dari setiap frame didapat, langkah selanjutnya adalah *Mel-Frequency Wrapping*. *Mel-Frequency Wrapping* merupakan langkah dimana frame sinyal pada domain frekuensi dirubah dalam bentuk mel frekuensi. Metode ini digunakan untuk melihat frame sinyal dalam bentuk histogram agar mudah dalam pengenalan frame. *Frequency wrapping* umumnya dilakukan dengan mengaplikasikan beberapa *filterbank*. *Filter* yang digunakan berjumlah 24 channel yang dilakukan secara linier terhadap frekuensi 0-4 kHz. Gambar 3.9 merupakan *mel-frequency filter bank* yang digunakan. Potongan *pseudocode* untuk *filterbank* dipresentasikan sebagai berikut.


```

for i := 0 to nMelFB +1 do
begin
    fPointFB[i]:=round(700*(exp((i*stepMelFB+k)/2595)-1));
end;
//hitung Mel Frekuensi Wrapping
for j := 1 to nMelFB do
begin
    for i := 0 to 4095 do
    begin
        if i<fPointFB[j-1] then
            melFB[j,i]:=0
        else if (i>=fPointFB[j-1]) And (i<fPointFB[j]) then
            melFB[j,i]:=(i-fPointFB[j-1])/
                (fPointFB[j]-fPointFB[j-1])
        else if (i>=fPointFB[j]) And (i<fPointFB[j+1]) then
            melFB[j,i]:=(fPointFB[j+1]-i)/
                (fPointFB[j+1]-fPointFB[j])
        else melFB[j,i]:=0;
    end;
end;
end;

```

Pemilihan frekuensi ini didasarkan pada frekuensi dimana suara manusia berada pada rentang ini. Keluaran dari proses ini adalah *filter bank* dalam *array* berjumlah 24 buah dengan masing-masing berisi koefisien penguatan beserta bandwidthnya. Kemudian untuk setiap *filter bank* yang dibentuk, maka akan dilakukan operasi perkalian dengan *magnitude* frekuensi sinyal suara. Potongan *pseudocode* untuk *mel-frequency wrapping* ditunjukkan sebagai berikut.

```

for j := 1 to nMelFB do
begin
    melPow[j]:=0;
    for i := 0 to panjang_fft div 2 -1 do
        melPow[j]:=melPow[j]+melFB[j,i]*xMag[i];
        melPow[j]:=power(melPow[j],0.33);
    end;
end;
for i := 1 to nMelFB do

```



```

begin
  xrr[i]:=0;
  xii[i]:=0;
  for j := 1 to nMelFB do
    begin
      xrr[i]:=xrr[i]+melPow[j]*cos(2*pi*(j)*(i)/nMelFB);
      xii[i]:=xii[i]-melPow[j]*sin(2*pi*(j)*(i)/nMelFB);
    end;
  cMel[i]:=(xrr[i]+xii[i])/nMelFB;
end;

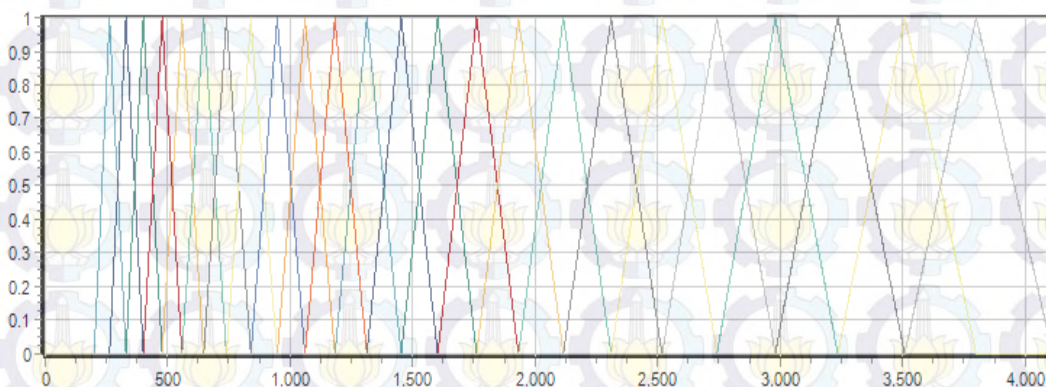
```

Cepstrum merupakan langkah dimana hasil *mel-frequency wrapping* dirubah kembali ke domain waktu. Metode yang digunakan adalah *discrete cosinus transform* (DCT). Potongan *pseudocode* dari DCT yang digunakan sebagai berikut.

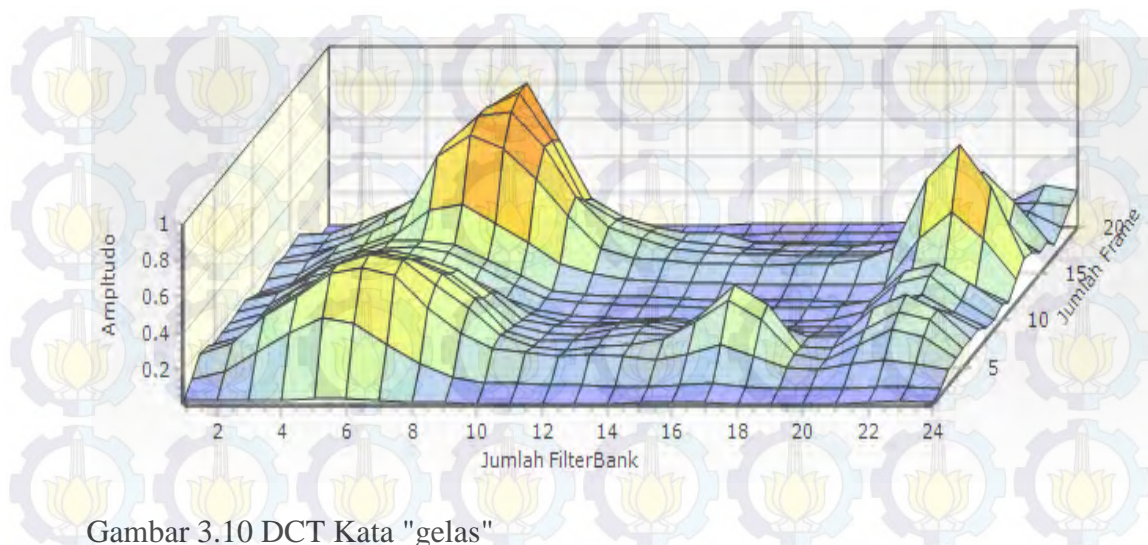
```

for i := 1 to nBarkFB do
  begin
    cBark[i]:=0;
    for j := 1 to nMelFB do
      cMel[i]:=cMel[i]+(melPow[j])*cos((i*pi*(j+0.5))/nMelFB);
    end;
  end;

```



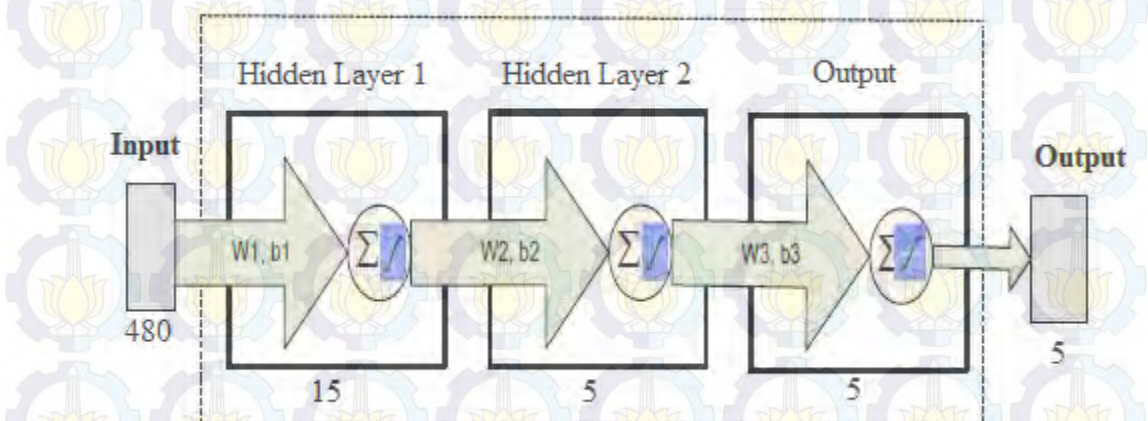
Gambar 3.9 Filter Bank dari Mel-Frequency Wrapping



Gambar 3.10 DCT Kata "gelas"

Hasil dari DCT disebut dengan *Mel-Frequency Cepstrum Coefficients* (MFCC) dimana berupa pola kata dari perintah suara. Deret pola dari kata perintah suara inilah yang akan digunakan sebagai inputan dari neural network. Gambar 3.10 merupakan DCT dari kata "gelas" yang terbentuk.

Salah satu metode untuk mengidentifikasi pola adalah dengan menggunakan *neural network*. Pola yang dihasilkan dari MFCC dilatih menggunakan algoritma *back-propagation*. *Neural-Network backpropagation* akan mengambil keputusan untuk target suara sesuai dengan yang diperintahkan. Nilai keluaran tertinggi akan merepresentasikan kosakata yang sesuai dengan sinyal *input*. Gambar 3.11 merupakan topologi dari *neural network* yang digunakan pada pembelajaran pola MFCC.



Gambar 3.11 Topologi *Neural Network*

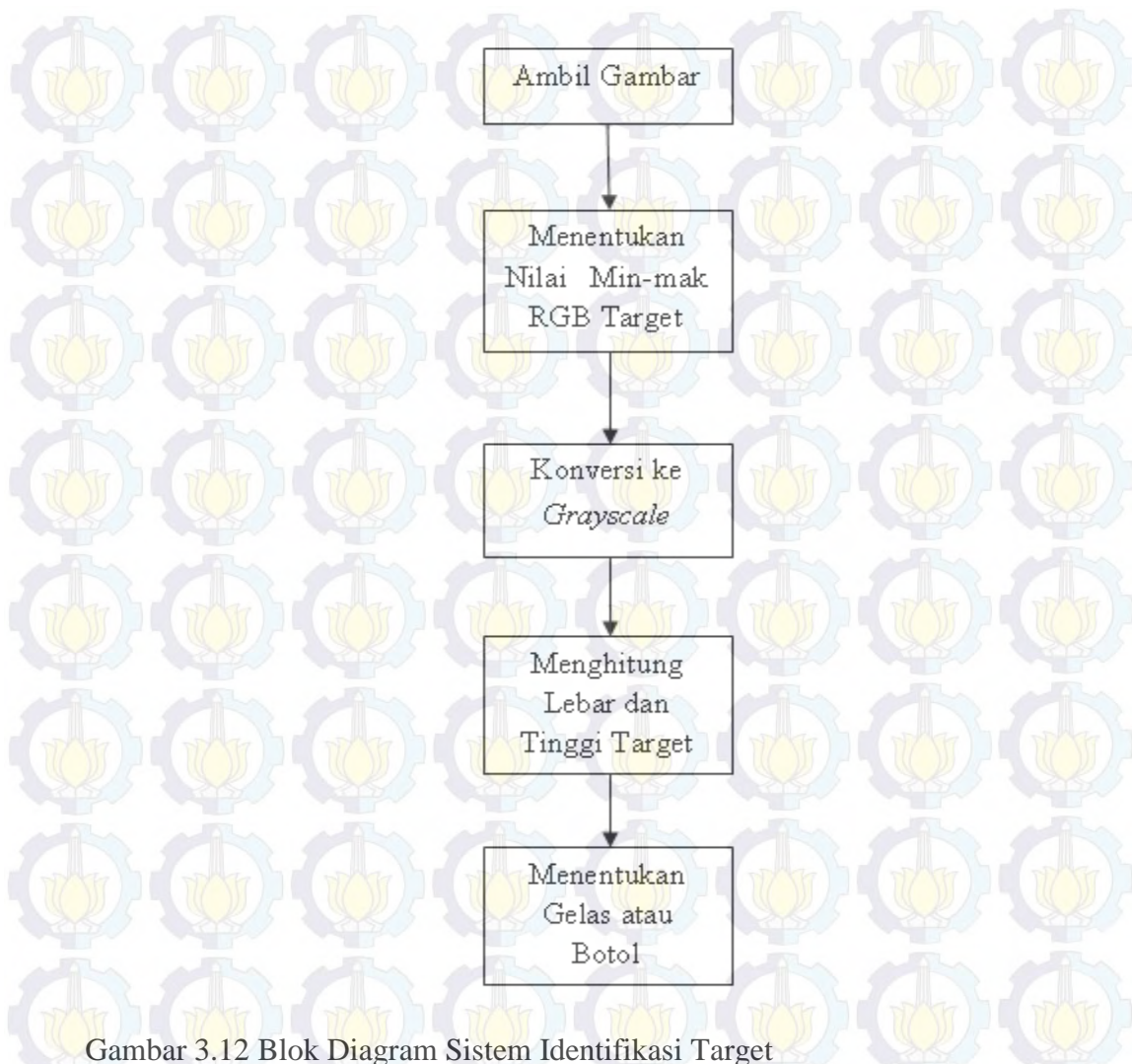
Tabel 3.1 Input-Output Pembelajaran NN

No	Input	Output				
		Y1	Y2	Y3	Y4	Y5
1	"gelas"	1	0	0	0	0
2	"botol"	0	1	0	0	0
3	"merah"	0	0	1	0	0
4	"biru"	0	0	0	1	0
5	"hijau"	0	0	0	0	1

Input terdiri dari pola kata yang dihasilkan dari MFCC, dimana setiap kata terdiri dari 480 data. NN yang digunakan memiliki 2 hidden layer dan fungsi aktivasi *sigmoid* untuk setiap *neuronnya*. Hidden layer pertama terdiri dari 15 node dan hidden layer kedua terdiri 5 node. Output terdiri dari lima yang merupakan kode biner sebagai perintah suara. Tabel 3.1 merupakan input-output pembelajaran dari *neural network*.

3.3 Sistem Identifikasi Target

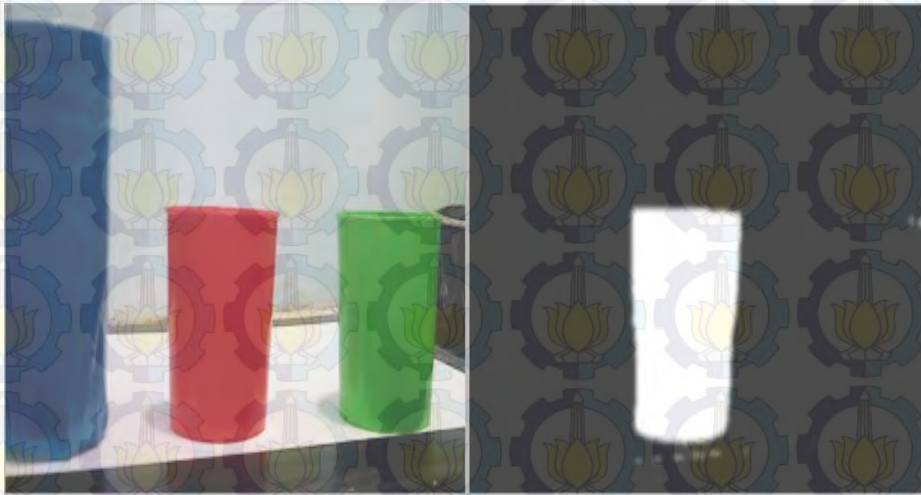
Sistem identifikasi target merupakan cara robot dapat mengenali objek yang menjadi target. Target terdiri dari gelas dan botol yang dibedakan warnanya. Target dikenali berdasarkan dua parameter. Parameter pertama target dikenali berdasarkan warna pada target. Parameter kedua target dikenali berdasarkan tinggi dan lebar target yang diambil kamera. Gambar 3.12 merupakan blok diagram sistem identifikasi target. Langkah pertama dalam identifikasi target yaitu mengambil gambar dari target. Gambar diambil menggunakan kamera robot. Gambar yang diambil berukuran 320x240 piksel. Gambar yang telah diambil diolah berdasarkan nilai minimum-maksimum dari tiap-tiap RGB. Warna target dapat diketahui dengan cara merubah-ubah nilai minimum-maksimum RGB dari gambar yang diambil. Perubahan nilai minimum-maksimum dilakukan secara manual hingga target yang dimaksud diketahui. Setelah nilai minimum-maksimum RGB dari tiap-tiap target didapatkan, langkah selanjutnya yaitu merubah gambar dalam bentuk *grayscale*.



Perpaduan nilai minimum-maksimum RGB yang telah dikonversi dalam bentuk grayscale akan membuat hanya warna target yang dikehendaki berwarna putih, sedangkan warna bukan target akan menjadi hitam. Sebagai contoh dalam mengenali warna merah pada gelas merah didapatkan nilai minimum-maksimum R adalah 150-256, G adalah 0-80, dan B adalah 0-80. Gambar 3.13 merupakan identifikasi warna merah dari gelas merah yang didapat. Berikut *pseudocode* identifikasi target berdasarkan warna

```

imgGrayColorFiltered = imgSmoothed.InRange(New Bgr(minBlue,
    minGreen,    minRed), New Bgr(maxBlue, maxGreen, maxRed))
imgGrayColorFiltered = imgGrayColorFiltered.PyrDown().PyrUp()
  
```

Gambar 3.13 Target Warna Merah dengan Nilai R(150-256), B(0-80), dan G(0-80)



Gambar 3.14 Identifikasi Lebar dan Tinggi Gelas Merah

Gambar dari target setelah dalam bentuk hitam-putih maka dapat dicari lebar dan tinggi dari target. Lebar dan tinggi dari target digunakan untuk membedakan target berupa gelas atau botol. Lebar dan tinggi dicari dengan cara menghitung jumlah piksel putih secara horizontal untuk lebar dan vertikal untuk tinggi target. Gambar 3.14 merupakan contoh dari identifikasi target berdasarkan lebar dan tinggi target. *Pseudocode* untuk mencari lebar dan tinggi target sebagai berikut.

```

flag = 0
For i = 0 To 239 - 1
  For j = 0 To 319 - 1
    If imgGrayColorFiltered2.Data(i, j, 0) = 255 Then

```



```

        y = i
        flag = 1
        Exit For
    End If
Next
    If flag = 1 Then
        Exit For
    End If
Next
flag = 0
For i = 0 To 319 - 1
    For j = 0 To 239 - 1
        If imgGrayColorFiltered2.Data(j, i, 0) = 255 Then
            x = i
            flag = 1
            Exit For
        End If
    Next
    If flag = 1 Then
        Exit For
    End If
Next
flag = 0

```

Menghitung lebar dan tinggi target berdasarkan piksel putih pada frame.

Pseudocode untuk menghitung lebar dan tinggi target sebagai berikut.

```

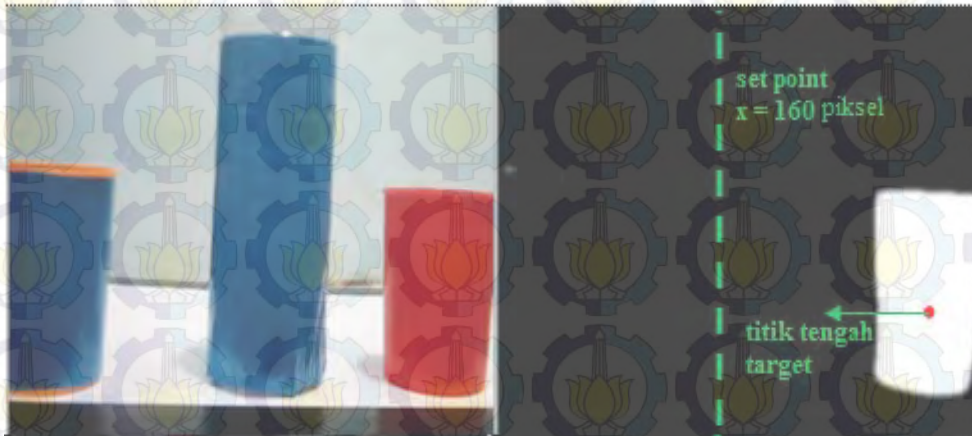
For i = 0 To 239 - 1
    For j = 0 To 319 - 1
        If imgGrayColorFiltered2.Data(239-i,319-j,0) = 255 Then
            tinggi = 239 - y - i
            flag = 1
            Exit For
        End If
    Next
    If flag = 1 Then
        Exit For
    End If
Next
flag = 0

For i = 0 To 319 - 1
    For j = 0 To 239 - 1
        If imgGrayColorFiltered2.Data(239-j,319-i,0)= 255 Then
            lebar = 319 - x - i
            flag = 1
            Exit For
        End If
    Next
    If flag = 1 Then
        Exit For
    End If
Next

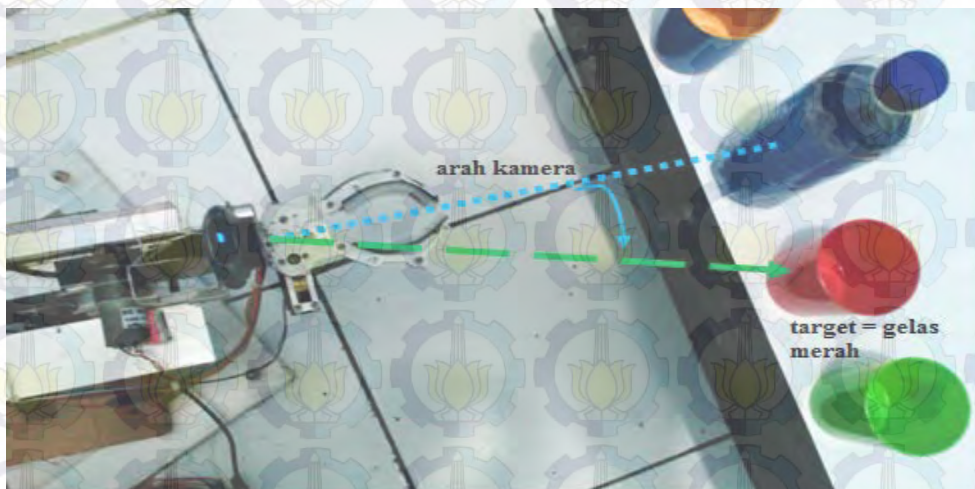
```


3.4 Sistem Kendali Robot Mencari Target

Sistem kendali robot mencari target merupakan cara robot memposisikan lengan robot sejajar dengan target yang diinginkan. Sensor yang digunakan berupa sensor kamera beresolusi 16 megapiksel. Frame gambar yang digunakan berukuran 320x240 piksel, sehingga set point terletak pada sumbu x piksel yang ke 160 atau setengah dari ukuran horizontal dari frame yang digunakan. Prosedur selanjutnya yaitu mencari titik tengah dari target. Titik tengah dari target didapat dari membagi dua lebar target. Gambar 3.15 menunjukkan posisi set point dan titik tengah dari gambar yang diambil kamera dengan target berupa gelas merah. Sedangkan Gambar 3.16 merupakan posisi lengan robot dan target tampak atas.



Gambar 3.15 Posisi Set Point dan Titik Tengah dari Target yang Diambil Kamera



Gambar 3.16 Posisi Set Point dan Posisi Target Tampak Atas

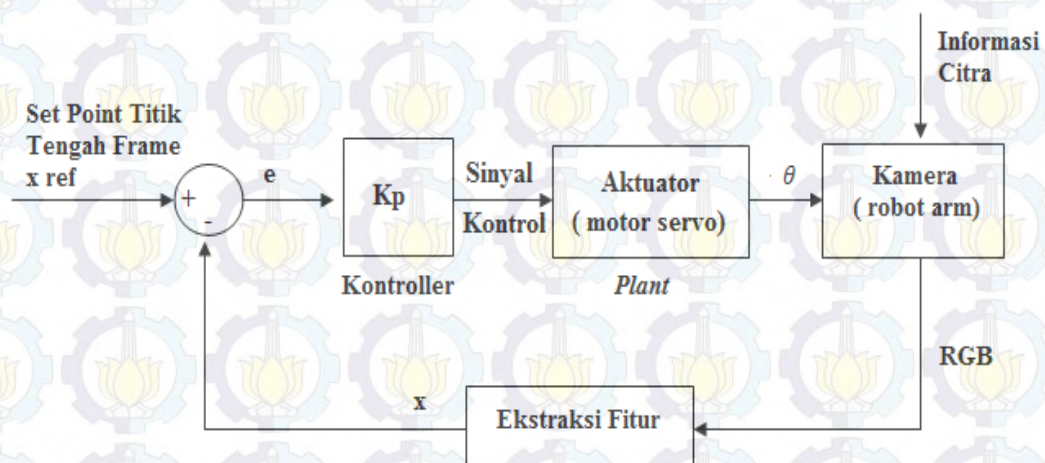
Robot mencari target dengan menggerakkan motor servo. Robot mencari target secara berputar sejauh 360°. Motor servo berputar ke kanan per satu derajat dalam mencari target. Kontrol propotional diterapkan pada robot dalam mencari target dengan toleransi error sebesar 10%. Error didapatkan dari selisih set point terhadap titik tengah target pada koordinat x. Error maksimum ketika target belum tertangkap kamera sebesar 160 piksel. Nilai Kp dicari secara manual atau coba-coba. Pengujian dilakukan untuk mendapatkan nilai Kp yang sesuai dengan desain kontrol. Ekstraksi fitur merupakan proses identifikasi target dalam menentukan nilai titik tengah target dalam koordinat x. Blok kontrol untuk sistem kendali robot mencari target ditunjukkan pada Gambar 3.17. Potongan *pseudocode* untuk propotional kontrol robot mencari target ditunjukkan sebagai berikut.

```

kp = 10
titik tengah target = x + lebar/2
error = 160 - titik tengah target
p = kp * error

```

Gambar 3.18 merupakan posisi tampak atas dari lengan robot ketika menemukan target. Sedangkan Gambar 3.19 merupakan gambar yang diambil dari kamera robot ketika lengan robot berada pada posisi target. Pada Gambar 3.19 terlihat posisi titik tengah target lebih 10 piksel dari posisi set point yang diharapkan.



Gambar 3.17 Blok Sistem Kendali Robot Mencari Target.



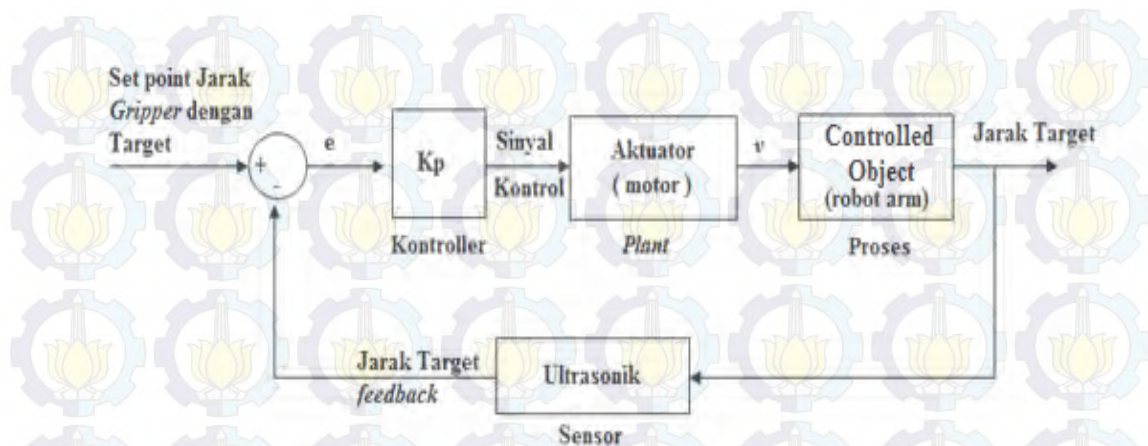
Gambar 3.18 Posisi Lengan Robot Terhadap Target yang Diinginkan



Gambar 3.19 Posisi Set Point Terhadap Titik Tengah Target Diambil Kamera

3.5 Sistem Kendali Robot Mengambil Target

Sistem kendali robot mengambil target merupakan cara robot memanjangkan lengan robot untuk mengambil target yang diinginkan. Sensor jarak yang digunakan berupa sensor ultrasonik. Kontrol proposional diterapkan pada robot dalam mengambil target dengan toleransi error sebesar 10%. Motor DC digunakan pada lengan robot untuk memanjang dalam mengambil target. Error dihasilkan berupa selisih set point terhadap hasil pembacaan jarak target oleh sensor. Set point yang diinginkan jarak target terhadap lengan robot berjarak 4cm. Nilai K_p yang digunakan didapatkan dengan cara *Ziegler-Nichols*.



Gambar 3.20 Blok Sistem Kendali Robot Mengambil Target

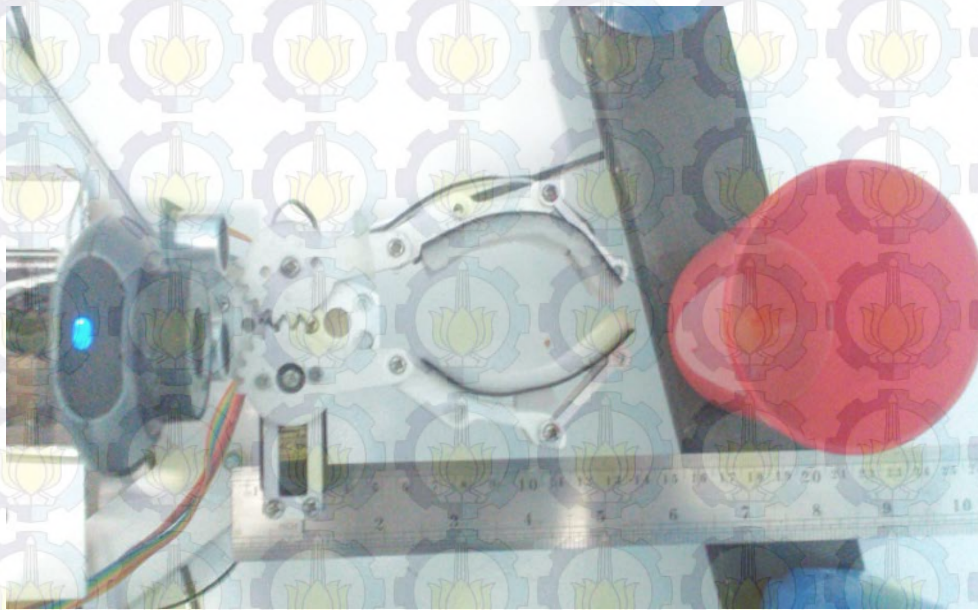
Keluaran dari motor berupa kecepatan dalam mengerjakan lengan robot memanjang dan memendek dalam mengambil target. Blok kontrol dalam sistem kendali robot mengambil target ditunjukkan pada Gambar 3.20. Potongan *pseudocode* untuk propotional control robot mengambil target ditunjukkan sebagai berikut.

```

err = 4-pembacaanSensor;
kp = T/L;
P = abs(kp*err);
if((err<0)&&(d!=0)){
    maju(P);
}
else if((err>0)&&(d!=0)) {
    mundur(P);
}
else{
    berhenti();
}

```

Proses kontrol objek pada blok kontrol Gambar 3.20 yang dimaksud adalah proses dimana lengan robot bergerak memanjang dan memendek. Kluaran dari proses ini berupa jarak ultrasonik terhadap target.



Gambar 3.21 Posisi Ultrasonok Pada Lengan Robot Terhadap Jarak Target

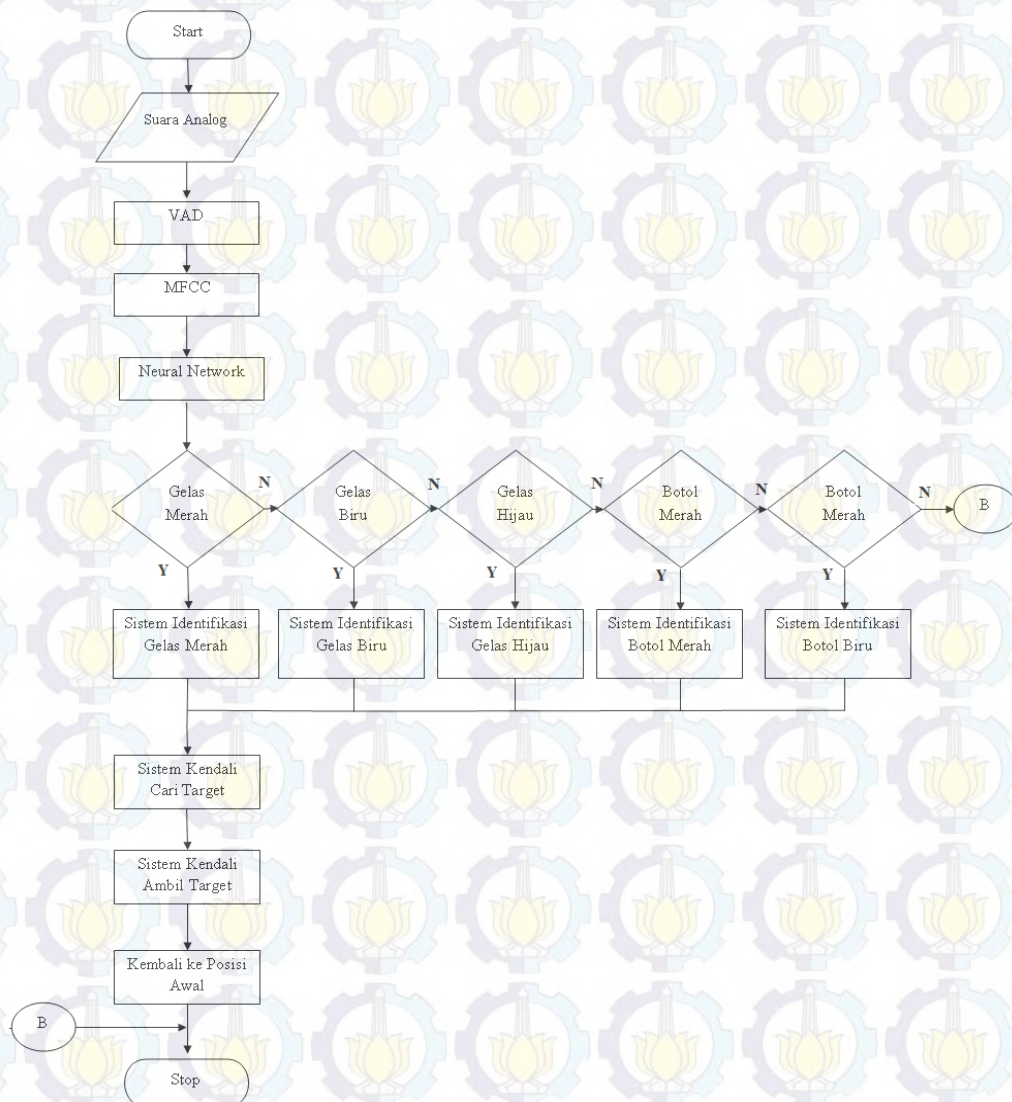


Gambar 3.22 Posisi Ultrasonik Pada Lengan Robot dengan Jarak Set Point Terhadap Target

Gambar 3.21 merupakan posisi lengan robot terhadap target dalam pembacaan jarak lengan robot terhadap target. Sedangkan Gambar 3.22 menunjukkan saat jarak lengan robot berada pada set point yang diinginkan.

3.6 Integrasi Perintah Suara dan Robot Lengan dalam Mengambil Target

Integrasi perintah suara dengan robot lengan merupakan pengambungan antara software dengan hardware robot. Software terdiri dari perintah suara dan pengenalan target, sedangkan hardware terdiri dari robot lengan tiga *degree of freedom*. Komunikasi yang digunakan dalam integrasi perintah suara dengan robot lengan menggunakan USB komunikasi serial. Gambar 3.23 merupakan *flowchart* integrasi perintah suara dengan robot lengan dalam mengambil target yang diperintahkan.



Gambar 3.23 Flowchart Integrasi Perintah Suara dengan Robot Lengan

Berdasarkan Gambar 3.23 masukan dari sistem berupa suara analog atau suara manusia. Suara direkam dengan menggunakan mikropon pada komputer. Proses awal dari suara terekam yaitu menentukan suara sebagai informasi, suara diam dan noise. Metode yang digunakan dalam menentukan suara informasi yaitu VAD. Setelah suara informasi yang berisi perintah suara terdeteksi langkah selanjutnya yaitu mengenali pola atau ciri dari suara informasi tersebut. Hal ini dilakukan untuk membedakan perintah suara yang diberikan. Metode yang digunakan untuk mengenali pola dari suara masukan pada penelitian ini menggunakan MFCC. Pola yang telah dikenali digunakan untuk data masukan *neural network*. *Neural network* digunakan sebagai pemberlajaran dari pola untuk membetuk suatu kode yang dipresentasikan sebagai perintah suara. Setelah perintah suara dikenali, sistem akan mengidentifikasi target sesuai dengan perintah suara yang diberikan. Setelah target dikenali atau teridentifikasi, sistem akan menggerakkan robot untuk mencari target. Sistem akan memerintahkan robot untuk mengambil target setelah target ditemukan. Proses akhir yaitu sistem memerintahkan robot untuk kembali ke posisi semula dengan membawa target. Namun apabila dalam proses pengenalan perintah suara tidak dikenali maka proses akan berakhir atau sistem tidak memerintahkan robot untuk bergerak.

BAB 4

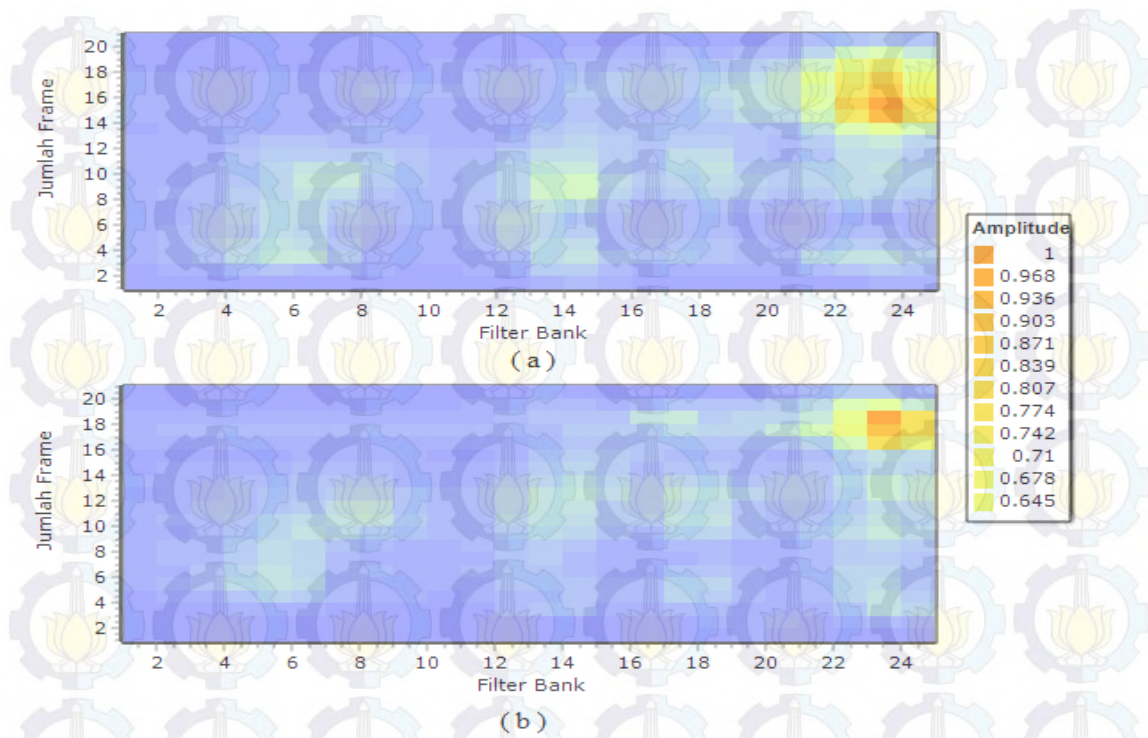
HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang hasil dari penelitian yang telah dilakukan. Hasil didapat dari pengujian-pengujian dari penelitian. Analisa dilakukan untuk mengetahui fungsional dari tiap-tiap pengujian. Pengujian yang dilakukan terdiri dari:

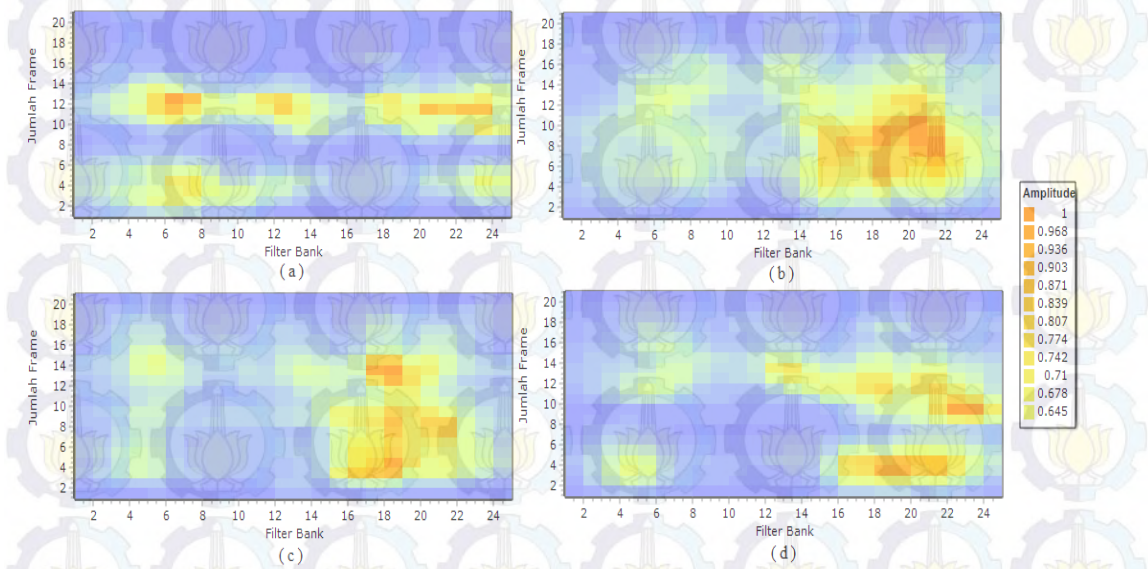
- ✓ Pengujian pembentukan pola perintah suara hasil dari MFCC.
- ✓ Pengujian sistem identifikasi pola dengan *neural network*.
- ✓ Pengujian sistem identifikasi target.
- ✓ Pengujian motor servo.
- ✓ Pengujian robot mencari target.
- ✓ Pengujian robot mengambil target.
- ✓ Pengujian robot mengambil target dengan perintah suara.

4.1. Pengujian Pembentukan Pola Perintah Suara Hasil Dari MFCC

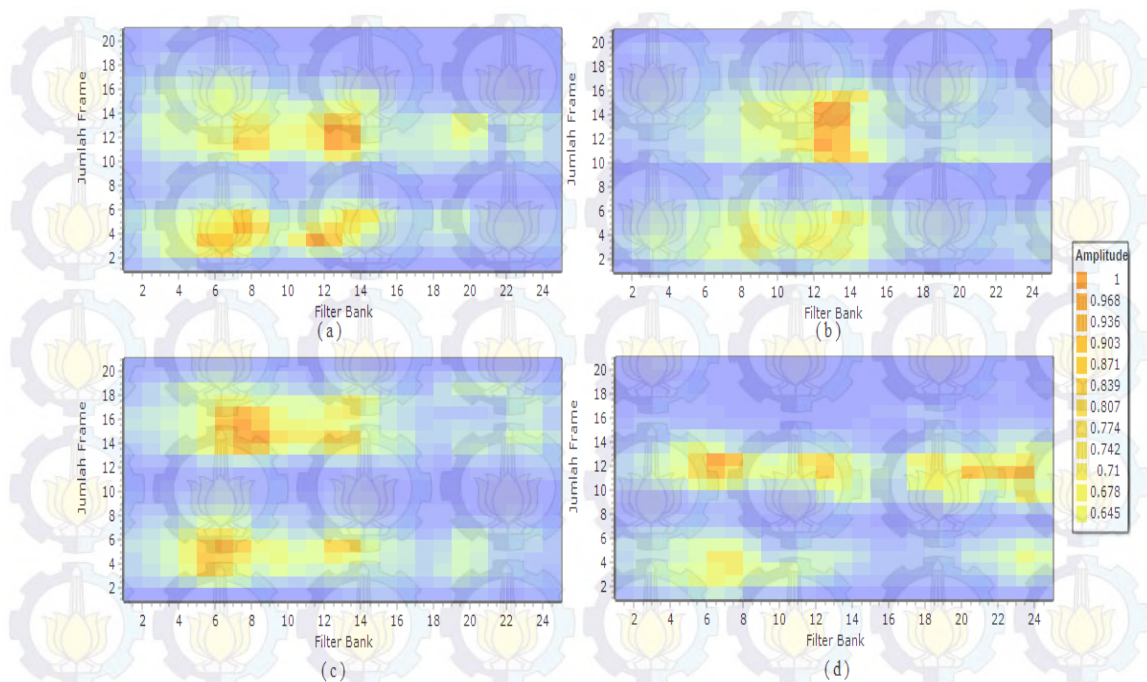
Pengujian pembentukan pola perintah suara hasil dari MFCC bertujuan untuk mengetahui pola dari tiap-tiap perintah suara yang dibentuk dari MFCC. Pola yang dibentuk terdiri dari 20 frame. Setiap frame memiliki 24 filter bank dengan amplitudo yang berbeda-beda. Pengujian dilakukan dengan cara melihat ciri atau pola yang dihasilkan dari MFCC dari tiap perintah suara. Pengujian pembentukan pola perintah suara terdiri dua jenis. Pengujian pertama dilakukan dengan mengucapkan perintah suara dengan kata yang sama dan pengujian kedua dilakukan dengan mengucapkan kata yang berbeda. Hasil dari ciri atau pola keluaran MFCC dari perintah suara kata "gelas" yang dilakukan sebanyak dua kali ditunjukkan pada Gambar 4.1. Gambar 4.1 menjelaskan bahwa pola yang dibentuk MFCC dari kata yang sama menghasilkan keluaran pola atau ciri yang menyerupai. Pengujian selanjutnya adalah dengan mengucapkan kata yang berbeda. Kata dari perintah suara yang digunakan yaitu "botol", "merah", "biru", dan "hijau". Hasil keluaran dari MFCC dengan mengucapkan kata yang berbeda ditunjukkan Gambar 4.2. Pola yang dihasilkan memiliki bentuk atau ciri yang berbeda dari tiap kata yang diujikan.



Gambar 4.1 Pola Keluaran MFCC dari Kata "gelas" (a) Pengucapan Pertama, (b) Pengucapan Kedua



Gambar 4.2 Pola Keluaran MFCC (a) Kata "botol", (b) Kata "merah", (c) Kata "biru", dan (d) Kata "hijau"



Gambar 4.3 Pola Keluaran MFCC Kata "botol" dari Subjek yang Berbeda-Beda

Pengujian juga dilakukan dengan subjek yang berbeda-beda. Gambar 4.3 merupakan hasil dari pola yang terbentuk dari kata "botol" dengan subjek yang berbeda-beda. Pola yang dibentuk memiliki pola yang berbeda dari subjek yang berbeda namun sedikit memiliki kemiripan. Sehingga memungkinkan perintah suara dari orang yang berbeda dapat dikenali. Pola-pola yang dihasilkan dari MFCC digunakan sebagai input *neural network* untuk diidentifikasi sebagai perintah suara.

4.2. Pengujian Sistem Identifikasi Pola dengan Neural Network

Pengujian sistem identifikasi pola dengan neural network bertujuan untuk mengetahui performa terbaik neural network yang telah dirancang. Pengujian dilakukan dengan 5 data inputan berupa suara analog sebagai target sebanyak sepuluh kali. Identifikasi performa terbaik dari neural network dilakukan dengan cara merubah-ubah nilai *learning rate* (μ), jumlah node pada hidden layer dan jumlah hidden layer yang digunakan. Performa terbaik dilihat dari persentase tingkat keberhasilan neural network dalam mengenali pola hasil dari MFCC. Hasil pengujian identifikasi pola dengan neural network ditunjukkan oleh Tabel 4.1

Tabel 4.1 Hasil Pengujian Identifikasi Pola dengan Neural Network

No	Hidden 1 (node)	Hidden 2 (node)	μ	MSE	Iterasi	Uji kata (20x)	Keberhasilan (%)
1	10	-	0.5	0.001	22600	gelas	85
						botol	80
						merah	90
						Biru	85
						hijau	80
2	15	-	0.5	0.001	21865	gelas	90
						botol	75
						merah	80
						biru	80
						hijau	75
3	15	5	0.5	0.001	21470	gelas	90
						botol	75
						merah	85
						biru	90
						hijau	85
4	15	5	0.8	0.001	20220	gelas	100
						botol	95
						merah	100
						biru	90
						hijau	90

Berdasarkan Tabel 4.1 performa *neural network* terbaik pada saat jumlah *node hidden layer* 1 sebanyak 15, jumlah *node hidden layer* 2 sebanyak 5 dan *learning rate* sebesar 0.8 dengan tingkat keberhasilan rata-rata sebesar 95%.

4.3. Pengujian Sistem Identifikasi Target

Pengujian sistem identifikasi target dilakukan untuk mengenali target yang akan diambil oleh robot lengan. Target diidentifikasi berdasarkan warna, lebar dan tinggi target yang diambil kamera.

4.3.1 Pengujian Identifikasi Target Berdasarkan Warna Target

Pengujian identifikasi target berdasarkan warna target bertujuan untuk mengenali warna target yaitu merah, biru atau hijau. Pengujian dilakukan dengan cara merubah-ubah nilai maksimum dan minimum RGB dari target. Pengujian identifikasi target berdasarkan warna dilakukan dua pengujian. Pengujian pertama dilakukan untuk mencari warna dari target dan pengujian kedua untuk membedakan target. Pengujian pertama dilakukan untuk mendapatkan nilai RGB yang sesuai dengan target. Sebagai contoh pengujian warna merah pada gelas merah. Pengujian mendeteksi gelas merah dilakukan dengan merubah-ubah nilai minimum dan maksimum nilai RGB hingga bentuk gelas merah dikenali. Gambar 4.4 menunjukkan pengujian mendeteksi gelas merah dengan nilai R minimum adalah 160 dan maksimum adalah 256, nilai G minimum adalah 0 dan nilai maksimum adalah 100, serta nilai B minimum adalah 0 dan nilai maksimum adalah 100.



Gambar 4.4 Target Warna Merah dengan Nilai R(160-256), B(0-100), dan G(0-100)



Gambar 4.5 Target Warna Merah dengan Nilai R(150-256), B(0-150), dan G(0-150)

Hasil dari pengujian terlihat gelas merah tidak dikenali secara sempurna. Pengujian selanjutnya dengan menurunkan nilai R minimum menjadi 150 dan nilai maksimum tetap 256, selanjutnya merubah nilai G dan B maksimum menjadi 150. Hasil dari pengujian terlihat gelas merah telah dikenali namun warna latar sebagian juga terkenali, seperti yang ditunjukkan pada Gambar 4.5 Pengujian selanjutnya dengan menurunkan nilai G dan B maksimum menjadi 80. Hasil dari pengujian gelas merah sudah dikenali dengan sempurna, seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Target Warna Merah dengan Nilai R(150-256), B(0-80), dan G(0-80)



Gambar 4.7 Target Warna Hijau dengan Nilai R(0-100), B(0-100), dan G(100-256)



Gambar 4.8 Target Warna Biru dengan Nilai R(0-20), B(50-256), dan G(0-80)

Setelah gelas merah dikenali, hal yang sama dilakukan untuk mengenali target yang lain. Hasil dari pengenalan target ditunjukkan pada Tabel 4.2

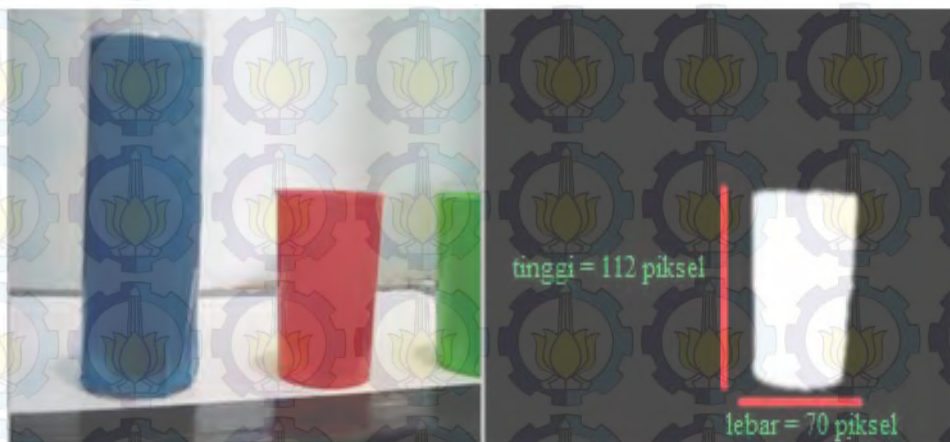
Tabel 4.2 Hasil Nilai Min-Mak Identifikasi Target Berdasarkan Warna

No	Target	R		G		B	
		Min	mak	min	mak	min	mak
1	Gelas Merah	150	256	0	80	0	80
2	Gelas Biru	0	20	0	80	50	256
3	Gelas Hijau	0	100	100	256	0	100
4	Botol Merah	150	256	0	80	0	80
5	Botol Biru	0	20	0	80	50	256

Tabel 4.2 menunjukkan perubahan-perubahan nilai minimum-maksimum RGB yang dilakukan dalam identifikasi target berdasarkan warna. Gambar 4.7 menunjukkan hasil dari pengenalan gelas hijau, dimana didapatkan nilai R dan B minimum sebesar 0 dan maksimum sebesar 100, sedangkan nilai G minimum sebesar 100 dan nilai maksimum sebesar 256. Untuk target botol biru didapatkan nilai R minimum sebesar 0 dan nilai maksimum sebesar 20, nilai G minimum sebesar 0 dan nilai maksimum sebesar 80, serta nilai B minimum sebesar 50 dan nilai maksimum sebesar 256. Pengenalan target botol biru ditunjukkan oleh Gambar 4.8.

4.3.2 Pengujian Identifikasi Target Berdasarkan Tinggi dan Lebar Target

Pengujian identifikasi target berdasarkan tinggi dan lebar target bertujuan untuk menentukan target berupa gelas atau botol. Pengujian dilakukan dengan cara menentukan terlebih dahulu target berdasarkan warnanya. Setelah target dikenali maka gambar yang diambil kamera dirubah dalam bentuk binari atau hitam putih. Target yang dikenali akan berwarna putih. Lebar dan tinggi target dapat diketahui dengan cara menghitung piksel warna putih yang dihasilkan. Lebar dan tinggi target dalam penelitian ini menggunakan dalam bentuk jumlah piksel. Gambar 4.9 menunjukkan hasil pengenalan target gelas merah dengan lebar 70 piksel dan tinggi 112 piksel. Sedangkan Gambar 4.10 merupakan hasil pengenalan target botol merah dengan lebar 65 piksel dan tinggi sebesar 180 piksel.



Gambar 4.9 Identifikasi Lebar dan Tinggi Gelas Merah



Gambar 4.10 Identifikasi Lebar dan Tinggi Botol Merah

Tabel 4.3 Hasil Identifikasi Target Berdasarkan Lebar dan Tinggi

No	Target	Jarak dari Kamera (cm)	Lebar (piksel)	Tinggi (piksel)
1	Gelas	15	152	124
		18	143	126
		20	132	128
		22	114	130
		25	70	112
2	Botol	15	68	176
		18	66	178
		20	65	180
		22	65	180
		25	65	180

Tabel 4.3 merupakan hasil dari identifikasi target berdasarkan lebar dan tinggi target yang diambil kamera. Jarak kamera terhadap target mempengaruhi terhadap lebar dan tinggi suatu target. Semakin jauh jarak target dengan kamera maka lebar dan tinggi target yang diambil kamera juga semakin kecil.

4.4 Pengujian Motor Servo

Pengujian motor servo bertujuan untuk mengetahui pengaruh masukan lebar pulsa terhadap keluaran berupa posisi atau sudut yang dihasilkan. Pengujian dilakukan dengan cara memberikan sinyal input berupa lebar pulsa ON dari 1ms hingga 2ms. Gambar 4.11 merupakan pengujian motor servo dengan lebar pulsa ON sebesar 1ms. Sudut yang dihasilkan dari masukan lebar pulsa ON 1ms sebesar 0° . Selanjutnya lebar pulsa ON ditambah pertahap sebesar 0.0056 ms. Pada saat lebar pulsa ON sebesar 1.5 ms sudut yang dihasilkan servo sebesar 90° . Posisi ini biasa disebut dengan posisi normal servo. Gambar 4.12 merupakan pengujian ketika servo diberi masukan lebar pulsa ON sebesar 1.5ms. Pengujian dilanjutkan hingga lebar pulsa ON sebesar 2ms. Pada saat lebar pulsa ON sebesar 2 ms, sudut yang dihasilkan motor servo sebesar 170° . Hal ini sesuai dengan data servo yang digunakan penelitian ini, bahwa servo memiliki sudut maksimal sebesar $\pm 180^\circ$. Gambar 4.13 merupakan hasil pengujian dengan masukan lebar pulsa ON sebesar 2 ms.



Gambar 4.11 Pengujian motor servo dengan lebar pulsa ON 1ms



Gambar 4.12 Pengujian motor servo dengan lebar pulsa ON 1.5ms



Gambar 4.13 Pengujian motor servo dengan lebar pulsa ON 2ms

4.5 Pengujian Kecepatan Motor DC

Pengujian kecepatan motor dc bertujuan untuk mengetahui pengaruh masukan PWM terhadap kecepatan motor dc yang digunakan. Pengujian dilakukan dengan cara memberi sinyal PWM berupa duty cycle pada motor dc. Pengujian dimulai dari duty cycle 0% hingga 100%. Respon keluaran kecepatan motor dc dihitung dalam putaran per-menit. Gambar 4.14 merupakan pengujian

kecepatan motor dc menggunakan rotari encoder. Tabel 4.4 merupakan hasil dari pengujian kecepatan motor dc.



Gambar 4.14 Pengujian Kecepatan Motor DC

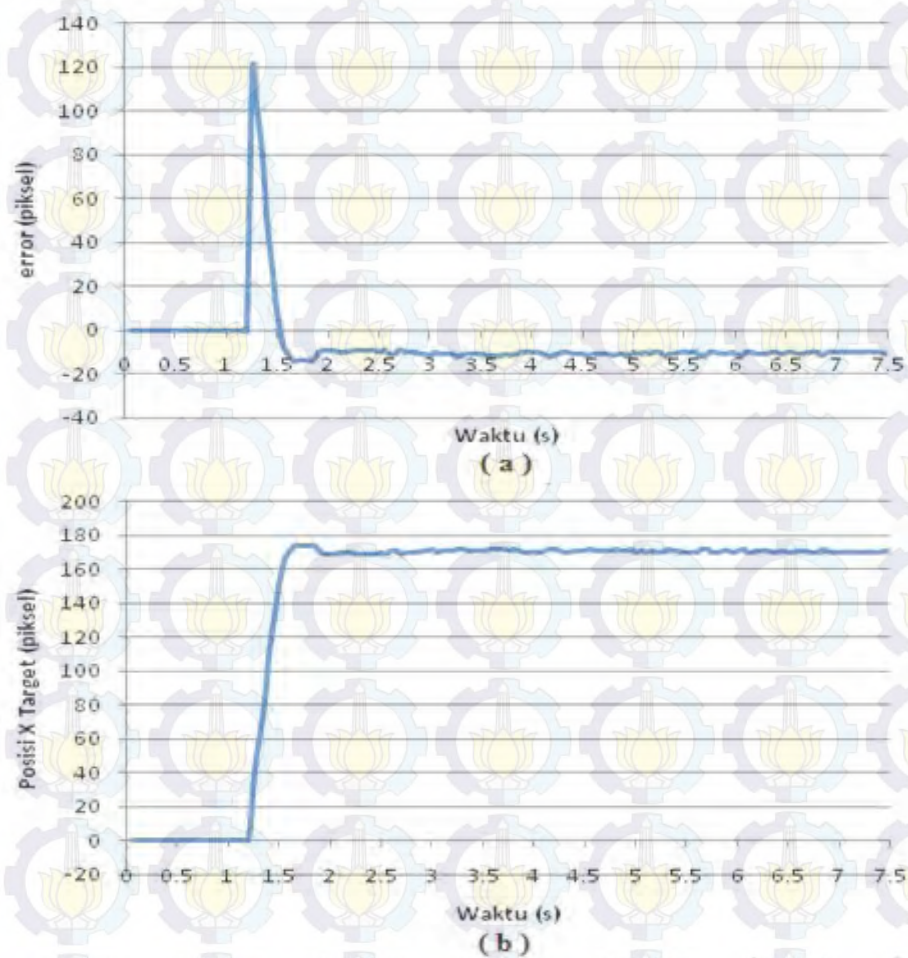
Tabel 4.4 Hasil Pengujian Kecepatan Motor DC

No	Duty Cycle (%)	Kecepatan (Rpm)
1	0	0
2	10	0
3	20	0
4	30	120
5	40	200
6	50	260
7	60	320
8	70	380
9	80	420
10	90	480
11	100	480

Berdasarkan Tabel 4.4 motor dc mulai berputar ketika diberi masukan duty cycle sebesar 33.33%. Kecepatan maksimum yang dihasilkan motor dc yaitu sebesar 480 Rpm. Hasil pengujian ini digunakan untuk parameter dalam menentukan sistem kontrol robot mengambil target.

4.6 Pengujian Robot Mencari Target

Pengujian robot mencari target memiliki tujuan mendapatkan respon parameter kontrol sesuai dengan desain yang diinginkan, yaitu toleransi error sebesar 10%. Prosedur pengujian dilakukan dengan cara merubah-rubah nilai Kp hingga mendapatkan nilai respon dengan toleransi error yang diinginkan. Tabel 4.5 merupakan hasil pengujian mendapatkan nilai Kp dari robot mencari target.



Gambar 4.15 Grafik Respon Parameter Kontrol Robot Mencari Target

Tabel 4.5 Hasil Pengujian Robot Mencari Target

No	Kp	Overshoot (piksel)	Settling Time (detik)	Error (piksel)	Error (%)
1	14	40	8	20	12.5
2	13	20	5	17	10.62
3	12	16	4	15	9.37
4	11	14	3	12	7.5
5	10	10	1	10	6.25
6	9	15	3	12	7.5

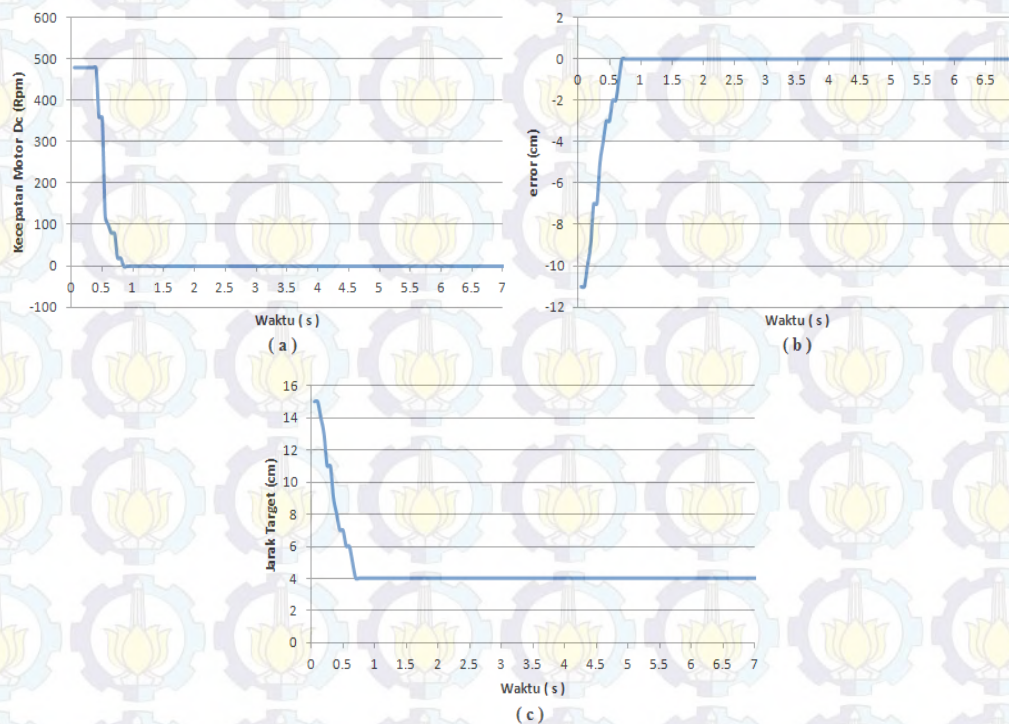
Berdasarkan hasil pengujian pada Tabel 4.5 didapatkan nilai Kp sebesar 10 dengan error yang dihasilkan sebesar 10 piksel. Hal ini sesuai dengan desain kontrol yang diinginkan yaitu toleransi error sebesar 10%. Gambar 4.15 merupakan grafik respon parameter kontrol yang dihasilkan dengan besar Kp adalah 10.

4.7 Pengujian Robot Mengambil Target

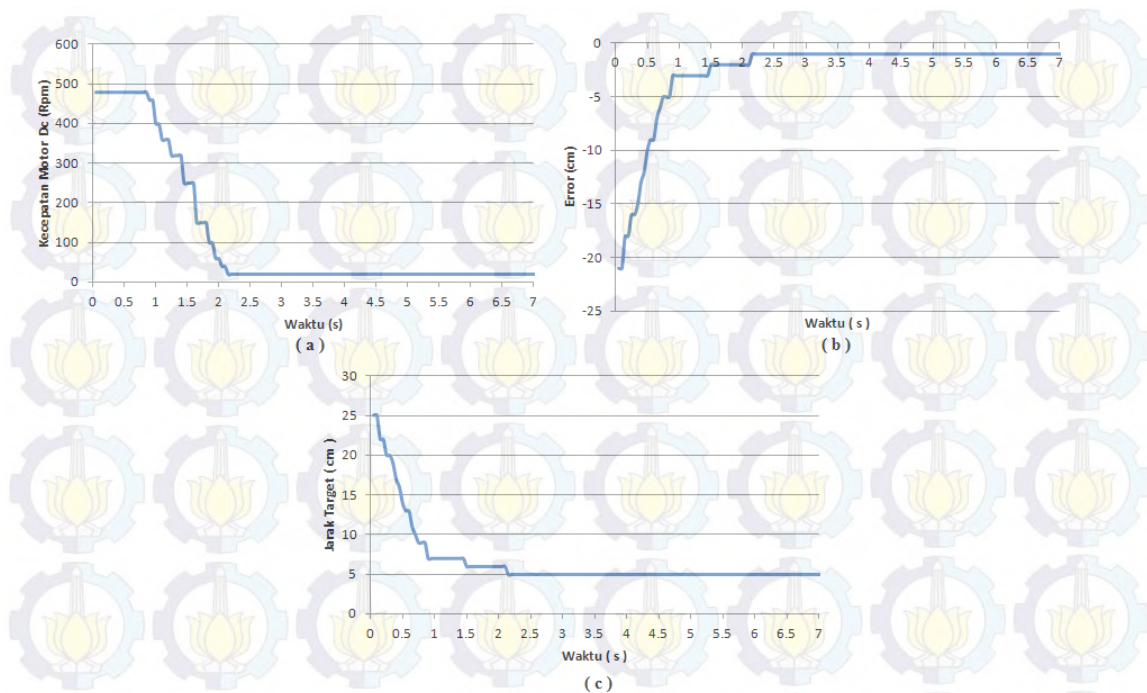
Pengujian robot mengambil target bertujuan memandu lengan robot dalam mengambil target dengan jarak target yang berbeda-beda. Pengujian dilakukan dengan meletakkan target pada jarak yang berbeda-beda. Set point jarak lengan robot dengan target sebesar 4 cm. Nilai Kp pada pengujian ini didapat dengan menggunakan zigler nichol. Tabel 4.6 merupakan hasil pengujian robot dalam mengambil target. Berdasarkan Tabel 4.6 robot memiliki tingkat keberhasilan yang tinggi dalam mengambil target. Hal ini direpresentasikan dari nilai error yang didapat ketika robot mengambil target. Settling time tertinggi terjadi pada saat jarak target berada pada jarak maksimum lengan yaitu 25 cm. Gambar 4.16 merupakan grafik respon parameter kontrol robot mengambil target dengan jarak target 15 cm. Sedangkan Gambar 4.17 merupakan grafik gesponp kontrol robot mengambil target dengan jarak target 25 cm.

Tabel 4.6 Hasil Pengujian Robot Mengambil Target

No	Nilai Kp	Jarak Target (cm)	Overshoot (cm)	Settling Time (detik)	Error (cm)	Error (%)
1	Kontrol P dengan Ziegler Nichol	15	0.1	1	0.1	0.5
2		15	0.2	0.6	0.2	1
3		15	0.1	1.4	0.1	0.5
4		16	0.1	0.9	0.3	1.5
5		18	0.2	1.1	0.5	2
6		20	0.1	1.3	1	4
7		22	0.1	1.5	1	4
8		25	0.1	2.1	1	4



Gambar 4.16 Grafik Respon Parameter Kontrol Robot Mengambil Target dengan Jarak Target 15 cm

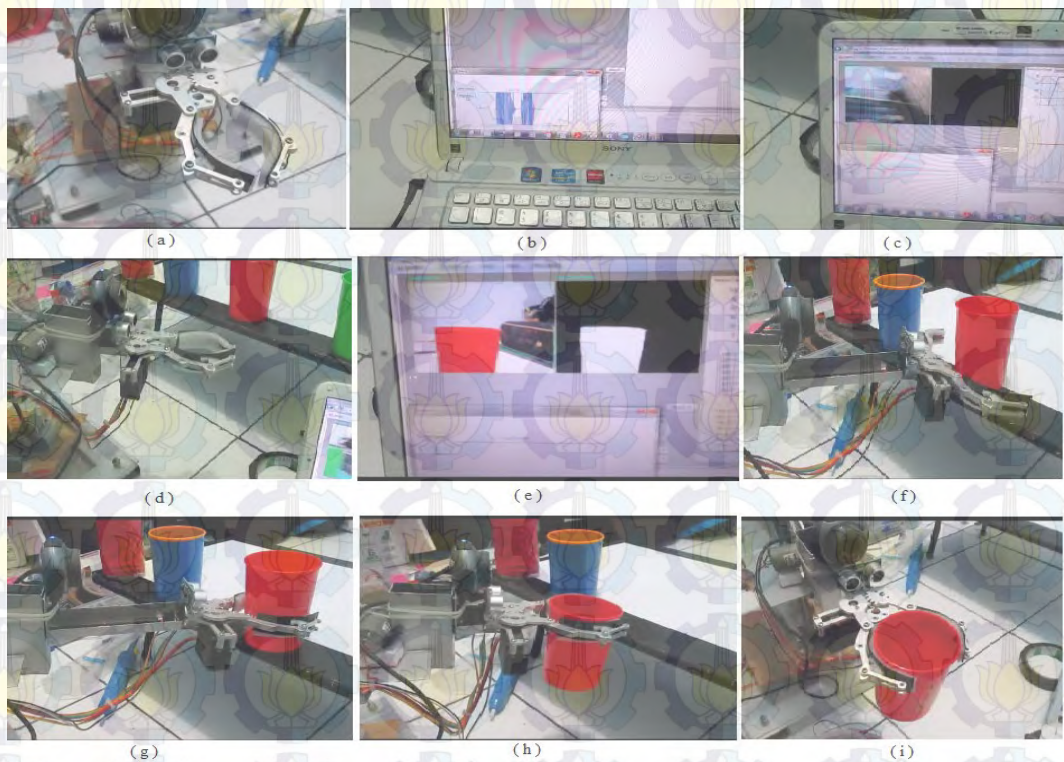


Gambar 4.17 Grafik Respon Parameter Kontrol Robot Mengambil Target dengan Jarak Target 25 cm

4.8 Pengujian Robot Mengambil Target dengan Perintah Suara

Pengujian robot mengambil target dengan perintah suara bertujuan untuk mengetahui berapa persen akurasi dari robot lengan dalam mengambil target menggunakan perintah suara. Pengujian dilakukan dengan cara memberikan perintah suara pada robot berupa nama target, kemudian melihat akurasi dari robot dalam mencari dan mengambil target sesuai dengan yang disebutkan. Gambar 4.18 merupakan proses salah satu pengujian robot mengambil gelas merah dengan menggunakan perintah suara. Gambar 4.18a merupakan kondisi awal robot dalam keadaan diam. Kemudian perintah suara dimasukkan pada komputer dengan menyebutkan nama target yaitu gelas merah, ditunjukkan oleh Gambar 4.18b. Pada Gambar 4.18c perintah suara telah dikenali dan kamera telah diaktifkan. Gambar 4.18d robot mulai bergerak berputar mencari target. Gambar 4.18e adalah ketika robot menemukan target yang dimaksud yaitu gelas merah. Setelah target ditemukan, *gripper* pada lengan robot akan membuka dan motor pada lengan robot akan menggerakkan lengan memanjang mendekati target. Sensor ultrasonik

memandu lengan robot dalam mengetahui jarak target. Gambar 4.18f merupakan posisi ketika jarak lengan robot berada pada set pint yaitu 4 cm dari jarak target. Kemudian *gripper* pada lengan robot akan menggenggam target hingga *limit switch* pada salah satu ujung *gripper* aktif. Setelah target telah digenggam robot akan mengambil target dengan cara memendekkan lengan robot, ditunjukkan pada Gambar 4.18h. Gambar 4.18i merupakan akhir dari proses dimana robot membawa target kembali pada posisi awal robot mulai. Pengujian tidak hanya dilakukan oleh subjek yang sama namun juga dilakukan terhadap subjek yang berbeda-beda. Subjek terdiri dari tiga berjenis kelamin laki-laki dan dua berjenis kelamin perempuan. Setiap subjek diujikan dengan target dan jarak target yang berbeda-beda. Penentuan target dan jarak target yang diujikan pada subjek dilakukan secara acak. Hal ini dilakukan untuk mengetahui pengaruh target yang diambil dan jarak target terhadap robot lengan. Subjek melakukan pengujian terhadap target sebanyak 5x. Akurasi diambil dari tingkat persen keberhasilan subjek dalam memerintahkan robot lengan mengambil target.



Gambar 4.18 Proses Pengujian Robot Mengambil Target dengan Perintah Suara

Tabel 4.7 Hasil Pengujian Robot Mengambil Target dengan Perintah Suara

No	Subjek	Jenis Kelamin	Target	Jarak Target (cm)	Keberhasilan (%)
1	1	P	Gelas Merah	15	100
2	2	L	Gelas Biru	20	80
3	3	L	Gelas Hijau	24	60
4	4	P	Botol Biru	15	80
5	5	L	Botol Hijau	20	60
6	2	L	Gelas Merah	15	100
7	3	L	Gelas Biru	20	80
8	4	P	Gelas Hijau	24	60
9	5	L	Botol Biru	15	80
10	2	L	Botol Hijau	20	80

Tabel 4.7 merupakan hasil yang didapat dari pengujian robot lengan dalam mengambil target dengan perintah suara. Dari Tabel 4.7 dapat diketahui bahwa jarak target juga mempengaruhi dari keberhasilan robot dalam mengambil target. Semakin jauh jarak target dengan robot semakin kecil tingkat keberhasilan robot mengambil target. Pengujian juga dilakukan dengan subjek yang berbeda-beda. Subjek terdiri dari laki-laki dan perempuan. Hasil dari subjek yang berbeda tidak mempengaruhi tingkat keberhasilan dari robot lengan mengambil benda menggunakan perintah suara. Secara keseluruhan robot lengan mengambil benda dengan perintah suara memiliki tingkat keberhasilan rata-rata sebesar 78 %.

BAB 5

KESIMPULAN DAN SARAN

Dari hasil perancangan, pembuatan, pengujian dan analisa sistem pada penelitian dengan judul “Robot Lengan Pengambil Benda Untuk Membantu Pasien Dengan Perintah Suara Menggunakan Metode MFCC dan *Neural Network*” ini, penulis mendapatkan kesimpulan serta memberikan saran yang akan berguna bagi pengembangan penelitian ini.

5.1 Kesimpulan

Pada penelitian yang telah dilakukan, metode MFCC mampu membentuk suara analog menjadi pola untuk dijadikan menjadi sebuah perintah suara. Pola yang dihasilkan oleh MFCC dengan kata dan subjek yang sama memiliki ciri yang menyerupai. Sedangkan pola yang dihasilkan dari kata yang berbeda dan subjek yang sama menghasilkan pola MFCC yang berbeda. Untuk kata yang sama dan subjek yang berbeda menghasilkan pola sedikit menyerupai. Sabagai identifikasi pola yang dihasilkan digunakan *neural-network*. *Neural network* dirancang sebagai pengambil keputusan dari pola yang dihasilkan untuk direpresentasikan sebagai perintah suara. Dari hasil pengujian *neural network* didapatkan performa terbaik *neural network* pada saat memiliki topologi dengan 15 *node hidden layer* pertama dan 5 *node hidden layer* kedua dengan *learning rate* sebesar 0.8. Target yang diambil dikenali dengan cara membedakan warna, lebar dan tinggi. Gerak robot mencari dan mengambil target memiliki respon terbaik pada saat nilai Kp sebesar 10. Pengujian secara keseluruhan didapatkan pengaruh jarak target terhadap akurasi robot dalam mengambil target. Semakin jauh jarak target yang diambil maka akurasi keberhasilan target dapat diambil semakin kecil. Secara rata-rata robot lengan pengambil benda dengan menggunakan perintah suara memiliki akurasi keberhasilan sebesar 78%.

5.2 Saran

Robot lengan yang digunakan pada penelitian ini tidak dapat berpindah posisi. Jangkauan robot dalam mengambil target hanya sebatas panjang maksimal dari lengan robot. Saran dari penulis untuk penelitian selanjutnya robot lengan yang digunakan dapat berpindah posisi sehingga memiliki jangkauan yang lebih luas

DAFTAR PUSTAKA

A. G. Buddhika P. Jayasekara, Keigo Watanabe, Kazuo Kiguchi, dan Kiyotaka Izumi. (2010), "Interpretation of Fuzzy Voice Commands for Robots Based on Vocal Cues Guided by User's Willingness", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan.

Adel Olabi, Richard Bearee, Olivier Gibaru, dan Mohamed Damak. (2010), "Feedrate planning for machining with industrial six-axis robots", *Control Engineering Practice*, Vol. 18 hal. 471-482.

Anand Mantri, Mukesh Tiwari, dan Jaikaran Singh. (2014), "Performance Evaluation of Human Voice Recognition System based on MFCC feature and HMM classifier", *Journal of Engineering Research and Applications*, ISSN : 2248-9622, Vol. 4, Issue 2(Version 1).

C. Bartneck, dan J. Forlizzi. (2004), "A Design-Centered Framework for Social Human-Robot Interaction," in *Proceedings of Ro-Man'2004*, Kura-shiki., hal. 591-594.

Elsen Ronando dan M. Isa Irawan. (2012), "Pengenalan Ucapan Kata Sebagai Pengendali Gerakan Robot Lengan Secara Real-Time dengan Metode Linear Predictive Coding – Neuro Fuzzy", *Jurnal Sains dan Seni ITS*, Vol. 1, No. 1, ISSN: 2301-928X.

Demuth, H. Beale, M. dan Hagan, M. (2009), *Neural Network Toolbox™ User's Guide*, The MathWorks, Inc.

Dietmar Knipp, Patrick G. Herzog, dan Helmut Stiebig. (2002), "Stacked Amorphous Silicon Color Sensors", *IEEE Transaction On Electron Devices*, VOL. 49, NO. 1.

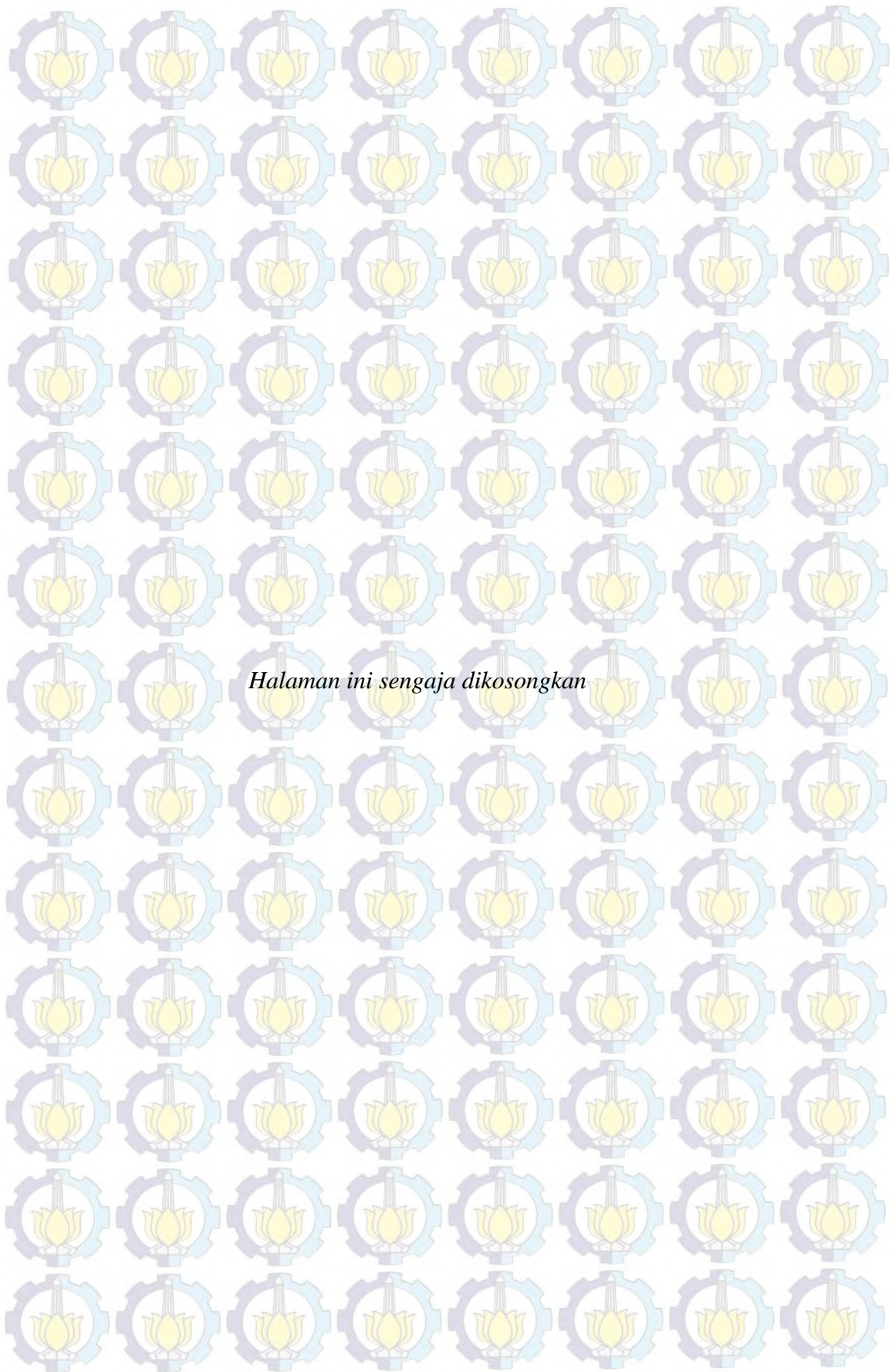
Gohil, V.J., Bhagwat, S.D., Raut, A.P., and Nirmal, P.R. (2013). "Robotics Arm Control Haptic Technology". *International Journal of Latest Research in Science and Technology*. Vol. 2, Issue 2. Hal. 98-102.

Gustica, A.N., Rivai, M., dan Tasripan. (2014). "Implementasi Sensor Gas pada Kontrol Lengan Robot untuk Mencari Sumber Gas". *Jurnal Teknik POMITS*. Vol. 3, No. 1. Hal. 90-95.

- J.Ramírez., Górriz, J.M., Segura, J.C., Puntonet, C.G., dan Rubio, A. (2006a), “Speech/Non-speech Discrimination based on Contextual Information Integrated Bispectrum LRT”, *IEEE Signal Processing Letters*, vol. 13, No. 8, pp. 497-500.
- J.Ramírez., Segura, J.C., Benítez, C., de la Torre, A., dan Rubio, A. (2005a), “An Effective Subband OSF-based VAD with Noise Reduction for Robust Speech Recognition”, *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 6, pp. 1119-1129.
- K. J. Parekh dan S. M. Sharma. (2013), “A Study on Kinematic Analysis of 3-DOF R-R-R Type Manipulator ARM”, *Indian Journal Of Applied Research*, Vol 3, Issue 13.
- Keigo Watanabe, dkk. (2006), “An Action Decision Mechanism Using Fuzzy-Neural Network in Voice Commanded Fuzzy Coach-Player System for Robots”, *SICE-ICASE International Joint Conference*, Korea.
- Kun Qian, Jie Niu, dan Hong Yang. (2013), “Developing a Gesture Based Remote Human-Robot Interaction System Using Kinec”, *International Journal of Smart Home*, Vol. 7, No. 4.
- Kshamamayee Dash, Debananda Padhi, Bhoomika Panda, dan Prof. Sanghamitra Mohanty. (2012), “Speaker Identification using Mel Frequency Cepstral Coefficient and BPNN”, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol 2, ISSN: 2277 128X.
- L. Rabiner dan B. H. Juang. (1993), “Fundamental Of Speech Recognition”, New Jersey.
- McAndrew Alasdair, (2004), *An Introduction to Digital Image Processing with Matlab. Notes for SCM2511 Image Processing 1*, School of Computer Science and Mathematics Victoria University of Technology.
- Rizal Ramdhani. (2009), “Kendali Gerak Interaktif Robot Mobile Berbasis Suara Ucapan”, Digilib ITS, Surabaya
- Saman Mousazadeh, dan Israel Cohen. (2006), “Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering”, *IEEE*, Vol. 21, NO.6.

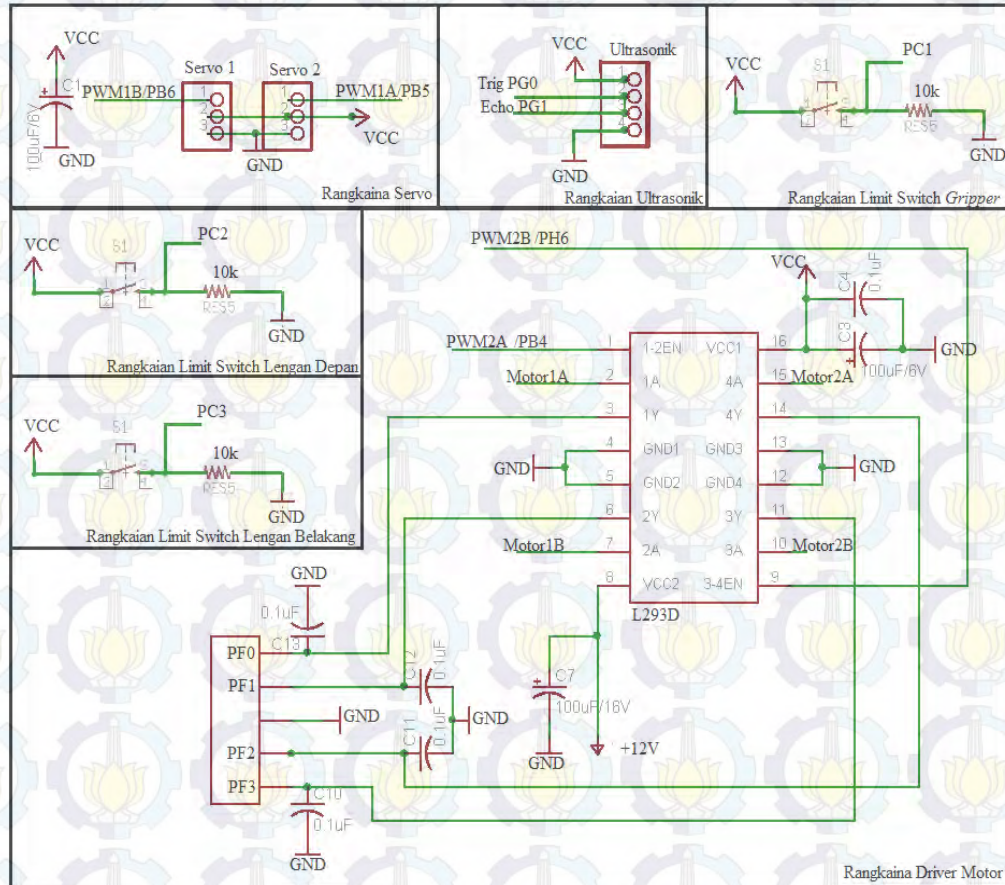
Soon-Hyuk Hong dan Jae Wook Jeon. (2001), "A Voice Command System for Autonomous Robots", *Transactions on Control Automation and Systems Engineering*, Vol. 3, No. .1.

UN. (2002). "United Nations and the International Federation of Robotics". *Proceedings of the World Robotics*, New York.

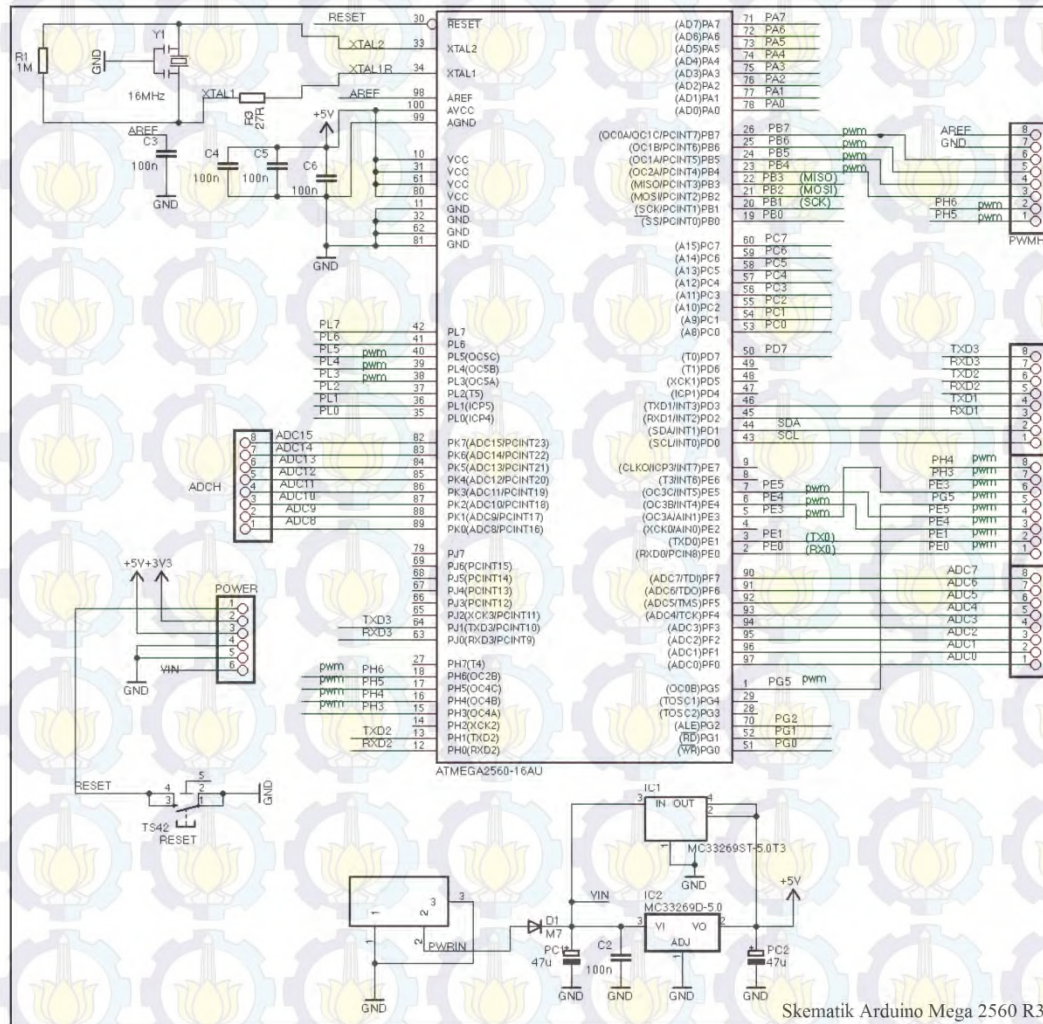


LAMPIRAN

Lampiran 1 Rangkaian Komponen Robot Lengan



Lampiran 2 Skematik Arduino Mega 250 R3



Skematik Arduino Mega 2560 R3

RIWAYAT HIDUP PENULIS



Wahyu Muldayani dilahirkan di Jember, 14 juni 1988. Penulis merupakan anak kedua dari tiga bersaudara pasangan Bapak Mulyono dan Ibu Marhani. Penulis memulai pendidikan di SD Negeri Kebonagung 01 Jember (1995-2001). Kemudian melanjutkan di SLTP Negeri 07 Jember (2001-2004). Penulis menempuh jenjang pendidikan selanjutnya di SMA Negeri 01 Glagah Banyuwangi (2004-2007). Pada tahun 2008 penulis diterima di

Program Sarjana Jurusan Teknik Elektro, Universitas Negeri Jember dan menyelesaikan studi sarjana pada tahun 2013. Setelah menyelesaikan studi tahap Sarjana, penulis meneruskan pendidikan di Program Magister Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember pada bidang keahlian Elektronika. Penulis telah melaksanakan sidang Tesis pada tanggal 7 Januari 2016. Penulis dapat dihubungi melalui email: wahyumuldayani@gmail.com.